



Informatică

Proba E. d)



Ghid pentru pregătirea examenului de BACALAUREAT 2020



10x2=20

Teste de antrenament
Rezolvări subiecte
Bareme de notare

- ✓ *Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică*
- ✓ *Filieră vocațională, profil militar, specializare matematică-informatică*
- ✓ *Filiera teoretică, profil real, specializare științele naturii*

Coordonatori,

prof. dr. **Genoveva Aurelia FARCAȘ**
INSPECTOR ȘCOLAR GENERAL

prof. **Mihaela Mariana ȚURA**
INSPECTOR ȘCOLAR GENERAL ADJUNCT

prof. **Emanuela-Tatiana PĂDURARIU**
INSPECTOR ȘCOLAR PENTRU INFORMATICĂ



Echipa de realizare a subiectelor și baremelor pentru testele de la disciplina **Informatică**

- prof. **ASOFEI Simina**, Liceul Teoretic "Miron Costin" Iași
- prof. **BUTNARAȘU Oana**, Liceul Teoretic de Informatică "Grigore Moisil" Iași
- prof. **CERCHEZ Emanuela**, Colegiul Național "Emil Racovita", Iași
- prof. **CREȚU Constantin**, Colegiul Național Iași
- prof. **CHELARIU Mihai**, Colegiul Național "Emil Racovita", Iași
- prof. **COȘNIȚĂ Otilia**, Liceul Teoretic "Miron Costin" Pașcani
- prof. **IVAȘC Cornelia**, Liceul Teoretic de Informatică "Grigore Moisil" Iași
- prof. **IUSCINSCHI Ingrid Simona**, Liceul Teoretic de Informatică "Grigore Moisil" Iași
- prof. **GRECU Silvia**, Liceul Teoretic de Informatică "Grigore Moisil" Iași
- prof. **GOREA ZAMFIR Claudiu-Cristian**, Colegiul Național Iași & Liceul Teoretic "Alexandru Ioan Cuza" Iași
- prof. **MAXIMIUC Ariadna**, Liceul Teoretic "Alexandru Ioan Cuza" Iași
- prof. **MĂCĂRESCU Roxana**, Liceul Teoretic "Dimitrie Cantemir" Iași
- prof. **MIRON Lucia**, Colegiul Național "C. Negruzzi" Iași
- prof. **MITRICĂ Eduard Gabriel**, Colegiul Național "Mihail Sadoveanu" Pașcani
- prof. **NEAGU Lucian**, Colegiul Național "C. Negruzzi" Iași
- prof. **NIȚĂ Mihai Daniel**, Colegiul Național "Garabet Ibrăileanu" Iași
- prof. **PETREA Tania**, Liceul Teoretic "Alexandru Ioan Cuza" Iași
- prof. **PĂDURARIU Emanuela-Tatiana**, Colegiul Economic Administrativ Iași
- prof. **ROȘCA Vasilica**, Colegiul Național "Garabet Ibrăileanu" Iași
- prof. **ROȘU Oana Alexandra**, Liceul Teoretic "Alexandru Ioan Cuza" Iași
- prof. **SCUTARU Sorina-Alina**, Liceul Teoretic "Miron Costin" Iași
- prof. **SUCIU Ionela**, Liceul Teoretic "Dimitrie Cantemir" Iași
- prof. **URICIUC Anca**, Liceul Teoretic "Dimitrie Cantemir" Iași
- prof. **VÂNGA Minodora Brîndușa**, Colegiul Național "Mihail Sadoveanu" Pașcani
- prof. **VRÂNCIANU Aurelia**, Liceul Tehnologic "Dimitrie Leonida" Iași
- prof. **ȚIBU Mirela**, Liceul Teoretic de Informatică "Grigore Moisil" Iași

Tehnoredactare computerizată:

- prof. **PĂDURARIU Emanuela-Tatiana**, Colegiul Economic Administrativ Iași
- prof. **VRÂNCIANU Aurelia**, Liceul Tehnologic "Dimitrie Leonida" Iași



ISBN 978-973-579-320-3

Casa Corpului Didactic "Spiru Haret" Iași

Str. Octav Botez 2 A, Iași, 700116

Telefon: 0232/210424; fax: 0232/210424

E-mail: ccdiasi@gmail.com, Web: www.ccdis.ro



Cuprins



Filiera teoretică, profil real, specializarea matematică-informatică / matematică-informatică intensiv informatică

Test 1	6
Rezolvare test 1	9
Barem test 1	12
Test 2	14
Rezolvare test 2	17
Barem test 2	20
Test 3	22
Rezolvare test 3	24
Barem test 3	26
Test 4	28
Rezolvare test 4	31
Barem test 4	33
Test 5	35
Rezolvare test 5	38
Barem test 5	42
Test 6	44
Rezolvare test 6	47
Barem test 6	51
Test 7	53
Rezolvare test 7	56
Barem test 7	59
Test 8	61
Rezolvare test 8	64
Barem test 8	67
Test 9	69
Rezolvare test 9	72
Barem test 9	75
Test 10	77
Rezolvare test 10	80
Barem test 10	83

Test 1	86
Rezolvare test 1	89
Barem test 1	92
Test 2	94
Rezolvare test 2	97
Barem test 2	100
Test 3	102
Rezolvare test 3	104
Barem test 3	106
Test 4	108
Rezolvare test 4	111
Barem test 4	113
Test 5	115
Rezolvare test 5	118
Barem test 5	122
Test 6	124
Rezolvare test 6	127
Barem test 6	130
Test 7	132
Rezolvare test 7	135
Barem test 7	138
Test 8	140
Rezolvare test 8	143
Barem test 8	145
Test 9	147
Rezolvare test 9	150
Barem test 9	153
Test 10	155
Rezolvare test 10	158
Barem test 10	162



INFORMATICĂ

Filiera teoretică, profil real, specializarea matematică-informatică / matematică-informatică intensiv informatică

Filiera vocațională, profil militar, specializarea matematică-informatică

MODEL TEST 1

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

Subiectul I
(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele **a**, **b** și **z** sunt întregi, iar **a** ≤ **b**. Care dintre expresiile C/C++ următoare are valoarea 1 dacă și numai dacă valoarea variabilei **z** este pară și nu aparține intervalului închis determinat de valorile variabilelor **a** și **b** ?

- a. **z%2==0 && z>a || z>b** b. **!(z<a && z>b) && z%2==0**
 c. **z<a && z>b && z%2==0** d. **!(z>=a && z<=b) && z%2==0**

2. Considerând declarația alăturată, în care câmpurile linia și coloana memorează poziția unei piese pe o tablă de șah, care dintre următoarele expresii are valoarea 1 dacă și numai dacă două piese **p1** și **p2** se află pe aceeași linie sau aceeași coloană?

```
struct { int linia, coloana; } p1, p2;
```

- a. **p1.linia==p2.linia && p1.coloana==p2.coloana**
 b. **p1.linia==p2.linia || p1.coloana==p2.coloana**
 c. **!(p1.linia!=p2.linia || p1.coloana!=p2.coloana)**
 d. **p1.coloana==p2.coloana**

3. Se consideră un arbore cu rădăcină cu 10 noduri, numerotate de la 1 la 10, memorat cu ajutorul vectorului de tați **T=(3,10,0,8,1,1,3,5,7,3)** care este lungimea maximă a unui lanț care unește rădăcina cu o frunză?

- a. 6 b. 2 c. 4 d. 5

4. Funcția **F** are definiția alăturată. Câte valori egale cu 0 se afișează la apelul **F(5)**?

```
void F(int x)
{   cout<<x<<' ';
    if(x!=0) {
        cout<<0<<' ';
        if (x%2==0)
            F(x-2);
        else
            F(x+1);
    }
}
```

- a. 9 b. 10 c. 5 d. 6

5. Se consideră algoritmul care determină toate permutările distincte de **n** obiecte (numerotate de la 1 la **n**), în care pe orice poziție de rang par se află o valoare pară. De exemplu, pentru **n=5**, primele trei permutări generate în ordine lexicografică sunt: (1,2,3,4,5), (1,2,5,4,3), (1,4,3,2,5). Pentru **n=5**, numărul total de astfel de permutări este:

- a. 12 b. 10 c. 9 d. 7

Subiectul al II-lea**(40 de puncte)****1. Se consideră algoritmul alăturat, descris în pseudocod.**

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

a) Scrieți valorile care se vor afișa dacă se citesc în ordine numerele **5 15 45 33 81 66 44 87**. (6p.)

b) Dacă pentru n, a și b se citesc valorile **5, 50, 100** completați setul de date cu valori care pot fi citite astfel încât, în urma executării algoritmului, valoarea afișată să fie **4**. (6p.)

c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

d) Scrieți în pseudocod un algoritm echivalent cu cel dat în care structura **pentru ...execută**

să fie înlocuită cu o structură repetitivă cu test inițial. (6p.)

citeste n, a, b (numere naturale nenule)

$m \leftarrow 0$

┌ pentru $i \leftarrow 1, n$ execută

| citește x

| ┌ dacă $x \geq a$ și $x \leq b$ atunci

|| ┌ dacă $x \% 10 = [x/10]$ atunci

|| | ┌ $m \leftarrow m + 1$

|| | └

|| └

└

scrie m

2. Ce se afișează după rularea următoarei secvențe de instrucțiuni? (6p.)

```
char s[50]="BAC informatica";
```

```
char *p,*q;
```

```
p= strtok(s, " ");
```

```
q= strtok(NULL, " ");
```

```
strcpy(s, strcat(q,p));
```

```
cout<<s;
```

3. Se consideră un graf neorientat cu 9 noduri, a cărui mulțime de muchii este $M = \{(1,2), (1,9), (2,3), (3,4), (3,7), (3,8), (4,5), (5,6), (5,7), (6,7), (6,8), (8,9)\}$. Indicați numărul minim de muchii și muchiile care pot fi eliminate astfel încât graful parțial obținut să fie eulerian și hamiltonian. (6p.)

Subiectul al- III-lea**(30 de puncte)**

1. O bucată de pânză bicoloră este reprezentată sub forma unui tablou bidimensional cu n linii și n coloane ce conține numere naturale din mulțimea $\{0,1\}$. Se numește submatrice de coordonate (i_1, j_1, i_2, j_2) o zonă dreptunghiulară din matrice care are colțul stânga-sus pe linia i_1 , coloana j_1 și colțul dreapta-jos pe linia i_2 , coloana j_2 ($i_1 \leq i_2, j_1 \leq j_2$). Dimensiunea unei submatrice este egală cu numărul de elemente din submatricea respectivă.

Spunem că pânza este perfectă dacă în matricea care o reprezintă există o submatrice de dimensiune maximă cu colțul stânga sus (i_1, j_1) și colțul dreapta jos (i_2, j_2) având toate elementele sale egale cu **0** și nu există elemente egale cu **0** în afara submatricei. Scrieți programul C/C++ care citește de la tastatură numărul natural n ($2 \leq n \leq 23$) și apoi $n \times n$ valori din mulțimea $\{0,1\}$ reprezentând elementele tabloului bidimensional și afișează pe ecran mesajul **DA**, dacă pânza este perfectă sau mesajul **NU** în caz contrar. (10p)

Exemplu: pentru $n=5$ și tabloul alăturat, se va afișa DA

```
1 1 1 1 1
```

```
1 0 0 0 1
```

```
1 0 0 0 1
```

```
1 1 1 1 1
```

```
1 1 1 1 1
```

Exemplu: pentru $n=5$ și tabloul alăturat, se va afișa NU

1 1 1 1 1

1 0 0 0 1

1 0 0 0 1

1 1 1 1 1

1 0 0 1 1

2. Un număr natural nenul se numește **echilibrat** dacă suma cifrelor de pe poziții pare este egală cu suma cifrelor de pe poziții impare, cifrele se numerotează de la dreapta la stânga începând cu valoarea 0. Exemplu: **121** este număr echilibrat pentru că $2=1+1$. Scrieți în C/C++ definiția completă a subprogramului **echilibrat**, cu doi parametri, a și b, prin care primește câte un număr natural ($2 \leq a < b \leq 10^9$, $b-a \leq 10000$). Subprogramul afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele **echilibrate** din intervalul **[a,b]**. Dacă în interval nu există astfel de numere, subprogramul afișează pe ecran mesajul **nu exista**. (10p)

Exemplu: pentru $a=100$ $b=150$, se afișează pe ecran: **143 132 121 110**.

3. O secvență de **K** elemente a unui șir de numere naturale este numită secvență **RK**, dacă elementele din secvență dau resturi distincte la împărțirea cu **K**.

Fișierul **bac.txt** conține pe prima linie un număr natural **K** din intervalul **[1,10]**, iar pe a doua linie conține un șir de cel puțin **K** și cel mult 10^5 numere naturale din intervalul **[0,10⁴]**, separate prin câte un spațiu.

Se cere să se afișeze pe ecran suma maximă a elementelor unei secvențe **RK** din șirul de pe a doua linie a fișierului, dacă în șir există secvențe **RK** sau mesajul **NU EXISTA** dacă șirul din a doua linie a fișierului nu conține nicio secvență **RK**.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplul 1. Dacă fișierul conține numerele

3

10 10 11 3 4 2 49 4 2 3 21 27 12 13 atunci se afișează pe ecran 24

(secvența **RK 10 11 3** are suma maximă)

Exemplul 2. Dacă fișierul conține numerele

3

10 11 13 16 11 10 atunci se afișează pe ecran **NU EXISTĂ** (în șir nu există nicio secvență **RK**).

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b) Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p)

Subiectul I

(20 de puncte)

1. d. $!(z \geq a \ \&\& \ z \leq b) \ \&\& \ z \% 2 == 0$
2. b. $p1.linia == p2.linia \ || \ p1.coloana == p2.coloana$
3. c. 4
4. c. 5
5. a. 12

Subiectul II

(40 de puncte)

1. a) 2 (6p.)

b) $n=5$ $a=50$ $b=100$ putem citi valorile 55 66 77 88 90 sau orice set format din 5 valori din care 4 sunt din mulțimea $\{55, 66, 77, 88, 99\}$ (6p.)

c) `#include <iostream>``using namespace std;``int main()``{ int n, a, b, i, x, m=0;``cin >> n >> a >> b;``for(i=1; i<=n; i++)``{ cin >> x;``if(x >= a && x <= b)``if(x % 10 == x / 10) m++;``}``cout << m;``return 0;``}`

d) structura repetitivă cu test initial este structura cat timp ... executa (6p.)

citește n, a, b (numere naturale nenule) $m \leftarrow 0; i \leftarrow 1$ ┌ cat timp $i \leq n$ execută| citește x | ┌ dacă $x \geq a$ și $x \leq b$ atunci|| ┌ dacă $x \% 10 = [x / 10]$ atunci|| | $m \leftarrow m + 1$

|| | └

|| └

| $i \leftarrow i + 1$

└

scrie m

2. informaticaBAC (6p.)

3. Numarul minim de muchii care pot fi eliminate este 3, muchiile care pot fi eliminate sunt $[3,7], [3,8], [5,6]$ (6p.)

Subiectul III**(30 de puncte)**

1. Vom determina în $i1, j1$ coordonatele primului element 0 din matrice și în $i2, j2$ coordonatele ultimului 0 din matrice, parcugem matrice și verificăm condițiile:

- elementele egale cu 0 sunt în submatricea determinate de cele 2 puncte și
- elementele egale cu 1 sunt în afara submatricei.

```
#include <iostream>
using namespace std;
int main()
{
    int n, a[25][25], i, j, i1, j1, i2, j2, ok=1, primul=0;
    cin >> n;
    for(i=1; i<=n; i++)
        for(j=1; j<=n; j++)
            { cin >> a[i][j];
              if(a[i][j]==0 && primul==0)
                  { i1=i; j1=j; primul=1; }
              if(a[i][j]==0)
                  { i2=i; j2=j; }
            }
    if (primul == 0) { cout << "NU"; return 0; }
    for(i=1; i<=n && ok; i++)
        for(j=1; j<=n && ok; j++)
            if(a[i][j]==1)
                { if(i>=i1 && i<=i2 && j>=j1 && j<=j2)
                    ok=0;
                }
            else
                { if(!(i>=i1 && i<=i2 && j>=j1 && j<=j2))
                    ok=0;
                }
    if(ok && primul) cout << "DA";
    else cout << "NU";
    return 0;
}
```

2. Căutăm numerele echilibrate numai printre numerele divizibile cu 11.

```
void echilibrat(int a, int b){
    int ok=0, si, sp, ci, r;
    for(int i=b; i>=a; i--){
        if(i%11==0){
            ci=i; si=0; sp=0; r=0;
            while(ci){ if(r%2==0)
                sp=sp+ci%10;
                else
                si=si+ci%10;
                ci=ci/10; r++; }
            if(si==sp){ cout << i << " "; ok=1; }
        }
    }
    if(!ok) cout << "nu exista";
}
```

3. a) O soluție eficientă utilizează un vector cu k elemente cu ajutorul căruia vom memora secvența curentă cu k elemente prin utilizarea vectorului circular, pentru a vedea dacă resturile la împărțirea cu k sunt distincte folosim un vector de apariții, dar și contorizăm câte resturi distincte avem în secvența curentă cu ajutorul variabilei nr , iar cu variabila s determinăm suma secvenței curente. La fiecare pas determinăm maximum acestor sume. Algoritmul este eficient din punct de vedere a spațiului de memorie deoarece utilizează doar variabile simple și 2 vectori a câte 10 elemente ($k \leq 10$). Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determină valoarea cerută, complexitatea algoritmului depinde doar de numărul de valori din fișier.

b)

```
#include <iostream>
#include<fstream>
using namespace std;
ifstream fin("bac.txt");
int k, i, x, v[10], s, smax,ap[10],nr,ok,y;
int main()
{  fin>>k;
   for(i=0;i<k;i++)
   {
       fin>>v[i];
       ap[v[i]%k]++;
       if(ap[v[i]%k]==1)nr++;
       s=s+v[i];
   }
   if(nr==k){smax=s;ok=1;}
   while(fin>>x)
   {
       y=v[i%k];s=s-y;
       if(ap[y%k]==1)nr--;
       ap[y%k]--;
       s=s+x;v[i%k]=x;ap[x%k]++;
       if(ap[x%k]==1)nr++;
       if(nr==k&& s>smax){smax=s;ok=1;}
       i++;
   }
   if(ok)cout<<smax;
   else cout<<"NU EXISTA";
   return 0;
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 1**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

1. d	2. b	3. c	4. c	5. a	5x4p.
-------------	-------------	-------------	-------------	-------------	--------------

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	6p.	Orice set format din 5 valori din care 4 sunt din mulțimea {55, 66, 77, 88, 99}
	c) Pentru program corect -declaraire variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă informaticaBAC	6p.	(*) Se acordă câte 2p. pentru fiecare aspect (determinarea fiecărui cuvânt, ordinea cuvintelor)
3.	Pentru rezolvare corectă Numarul minim de muchii care pot fi eliminate este 3 , muchiile care pot fi eliminate sunt [3,7],[3,8],[5,6] (*)	6p.	(*) Se acordă câte 3 puncte pentru fiecare aspect al cerinței. Dacă doar două muchii sunt corecte, se acordă doar 2 puncte din cele 3

SUBIECTUL al III - lea**(30 de puncte)**

1.	Pentru program corect -declararea variabilelor -citire a datelor - verificarea condițiilor cerute (*) -corectitudine a globală a programului ¹⁾	10p. 1p. 2p. 6p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (determinarea coordonatelor submatricei, valorile 0 în submatrice, valorile 1 în afara submatricei).
2.	Pentru subprogram corect -antet subprogram (*) -determinare a numărului cerut (**) - verificarea existenței valorilor cu proprietatea cerută -declarare a variabilelor locale, -corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (testarea paritatii unei cifre, calcularea sumelor, verificarea poziției în număr).
3.	a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizarea unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție eficientă utilizează un vector cu k elemente cu ajutorul căruia vom memora secvența curentă cu k elemente prin utilizarea vectorului circular, pentru a vedea dacă resturile la împărțirea cu k sunt distincte folosim un vector de apariții, dar și contorizăm câte resturi distincte avem în secvența curentă cu ajutorul variabile nr, iar cu variabila s determinăm suma secvenței curente. La fiecare pas determinăm maximul acestor sume. Algoritmul este eficient din punct de vedere a spațiului de memorie deoarece utilizează doar variabile simple și 2 vectori a câte 10 elemente ($k \leq 10$). Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determină valoarea cerută, complexitatea algoritmului depinde doar de numărul de valori din fișier.

MODEL TEST 2

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I
(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Stabiliți care dintre expresiile următoare are valoarea 1 dacă și numai dacă variabila reală x aparține reuniunii intervalelor $[2,10]$ și $(30,40)$?

- a) $!(x < 2 \parallel x > 10) \parallel (x \geq 40 \parallel x \leq 30)$
 b) $(x \leq 10 \ \&\& \ x \geq 2) \ \&\& \ (x > 30 \ \&\& \ x < 40)$
 c) $(x > 10 \ \&\& \ x > 2) \ \&\& \ (x \leq 30 \ \&\& \ x \leq 40)$
 d) $!(x < 2 \parallel x > 10) \parallel !(x \geq 40 \parallel x \leq 30)$

2. Ce valoare are apelul `sir(s,0)` al funcției următoare, dacă șirul de caractere `s` este `a1de2d33bc`?

```
int sir(char s[],int i){
  if (i<strlen(s)-1)
    if (s[i]>='a' && s[i]<='z' && s[i+1]<='9' && s[i+1]>='0')
      return 1+sir(s,i+1);
    else
      return sir(s,i+1);
  else return 0;
}
```

- a. 6 b. 5 c. 3 d. 4

3. Un graf neorientat are 7 muchii și conține două componente conexe. Fiecare componentă conexă din graf este arbore. Câte noduri are acest graf?

- a. 8 b. 10 c. 9 d. 11

4. Utilizând metoda backtracking, se generează, toate modalitățile de scriere a unui număr natural mai mare decât 1, ca sumă de numere prime, în ordine crescătoare. Pentru numărul 7, se generează în ordine combinațiile: 2+2+3, 2+5, 7. Utilizând același algoritm se generează toate modalitățile de scriere a numărului 18 ca sumă de numere prime, în ordine crescătoare. Care este a șasea soluție generată?

- a. 2+2+2+2+5+5 b. 2+2+2+3+3+3+3 c. 2+2+7+7 d. 2+2+2+5+7

5. Un graf neorientat are 7 noduri și este reprezentat prin listele de adiacență alăturate. Care dintre afirmațiile următoare este adevărată pentru acest graf?

- Graful este hamiltonian și eulerian
- Există un subgraf hamiltonian și eulerian format din 4 noduri al grafului dat
- Graful este complet
- Există un subgraf complet format din 4 noduri al grafului dat

1: 2,3,4,5

2: 1,3,4

3: 1,2,5

4: 1,2,5

5: 1,3,4

SUBIECTUL a II-lea**(40 de puncte)****1. Se consideră algoritmul alăturat descris în pseudocod:**

S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

a) Ce se va afișa, dacă se citesc, în această ordine, numerele **5,3,2324,31,7229,361,75** (6p.).

b) Scrieți un set de date de intrare care să determine afișarea valorii **200**.

(6p.)

c) Scrieți programul C/C++ corespunzător algoritmului dat.

(10p.)

d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **pentru** cu alt tip de structură repetitivă.

(6p.)

citește n, c (n, c numere naturale nenule, $c < 10$)

 $e \leftarrow 0$ pentru $i \leftarrow 1, n$ execută

citește x (număr natural nenul)

 $p \leftarrow 1$ $nr \leftarrow 0$ $cx \leftarrow x$ cat timp $x \neq 0$ executadaca $x \% 10 \leq c$ atunci $nr \leftarrow nr + (x \% 10) * p$ $p \leftarrow p * 10$ $x \leftarrow [x/10]$ daca $nr \% 2 = 0$ și $nr > 0$ atunci $e \leftarrow e + cx$ scrie e

2. Se consideră declarațiile de mai jos, în care tabloul unidimensional c memorează informații despre 200 de concurenți participanți la un concurs sportiv. Pentru fiecare concurent se memorează: numele prin câmpul **nume_concurent**, denumirea probei la care participă prin câmpul **proba** și punctajul obținut prin câmpul **punctaj**. Denumirea unei probe poate conține mai multe cuvinte separate prin spațiu. Variabila p memorează denumirea unei probe din concurs.

```
struct concurs{
    char nume_concurent[25],proba[50];
    int punctaj;
} c[200];
char p[50];
```

Scrieți o secvență de instrucțiuni C/C++ care citește variabila p , determină și afișează pe ecran, separate prin virgulă, numele concurenților care au obținut un punctaj egal cu 100 la proba cu denumirea p . (6p.)

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 0 la 4. Fără a utiliza alte variabile decât cele menționate, scrieți în limbajul C/C++, secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma execuției secvenței obținute, variabila a să memoreze tabloul alăturat.

```
for (i=0; i<5; i++)
  for (j=0; j<5; j++)
    .....
```

0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2
4	0	1	2	3

SUBIECTUL al III-lea**(30 de puncte)**

1. Subprogramul **perechi** primește prin parametrul n un număr natural ($2 \leq n \leq 100$), prin parametrul v primește un tablou unidimensional, format din n numere naturale nenule și distincte, fiecare număr având cel mult 9 cifre și prin parametrul p furnizează numărul de perechi distincte formate din elemente din tabloul v , care sunt prime între ele. Două numere naturale sunt prime între ele dacă au un singur divizor comun: numărul 1. Două perechi de elemente sunt distincte dacă diferă prin cel puțin un element. Scrieți în limbajul C/C++ definiția completă a subprogramului **perechi**.

Exemplu: dacă $n=6$ și $v=(4,22,12,9,63,28)$ atunci, după apel, se obține $p=5$, deoarece vectorul are 5 perechi de numere ce verifică proprietatea dată: $(4,9);(4,63);(22,9);(22,63);(9,28)$. **(10p.)**

2. Se consideră un număr natural nenul $n(n \leq 30)$, un cuvânt c , format din cel mult 20 de litere mici și un text format din n linii. Fiecare linie din text are cel mult 200 de caractere care sunt: litere din alfabetul englez, spații și se încheie cu caracterul newline. Fiecare linie din text este formată din mai multe cuvinte, separate între ele printr-un spațiu. Datele se citesc de la tastatură astfel: de pe prima linie se citește numărul n și cuvântul c , separate printr-un spațiu, de pe fiecare dintre următoarele n linii se citește câte o linie din text. Scrieți un program C/C++ care citește datele menționate, determină și afișează pe ecran pentru fiecare linie din text numărul de cuvinte ce au ca prefix cuvântul c , fără a face diferență între litere mari și litere mici. Cele n numere naturale afișate, corespunzător celor n linii din text, vor fi separate printr-un spațiu. De exemplu, dacă n este 4, cuvântul c este **cod** și textul este

Se poate cauta **codul** postal al unei locuințe și **codurile** postale ale locuințelor unei strazi
 Ai **codificat** și decodificat corect textul dat
 Adriana a recitat **Codul** Muncii **Codul** fiscal și **Codul** Administrativ
Codrin a mers la medic unde a primit un **cod** de diagnostic și apoi a recodat un modul din monocrom

programul va afișa **2 1 3 2**, deoarece prima linie are 2 cuvinte ce au ca prefix cuvântul **cod**, a doua linie are un cuvânt ce are ca prefix cuvântul **cod**, a treia linie are 3 cuvinte ce au ca prefix cuvântul **cod** și a patra linie are 2 cuvinte ce are ca prefix cuvântul **cod**. **(10p.)**

3. Fișierul BAC.TXT conține pe prima linie două numere naturale nenule n și m ($n \leq 1000$, $m \leq 10000$). A doua linie din fișier conține un șir a având n numere naturale, formate din cel mult trei cifre, în ordine strict crescătoare și separate prin câte un spațiu. A treia linie din fișier conține un șir b având m numere naturale, formate din cel mult trei cifre și separate prin câte un spațiu. Scrieți un program C/C++ care citește valorile din fișierul BAC.TXT și afișează pe ecran, în ordine crescătoare și separate prin spațiu, numerele din șirul b ce ar putea fi inserate în șirul a , astfel încât a să rămână ordonat strict crescător sau afișează mesajul Nu exista numere care vor fi inserate, dacă niciun număr din șirul b nu poate fi inserat în șirul a cu proprietatea de mai sus.

Se va utiliza un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de execuție. De exemplu, dacă fișierul BAC.TXT are conținutul următor, pe ecran se va afișa **3 35 90**.

6 7
 5 14 25 80 100 150
 5 3 90 14 35 3 35

- a. Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(2p.)**
 b. Scrieți programul C/C++ corespunzător algoritmului descris la punctul a. **(8p.)**

SUBIECTUL I

1. d 2. c 3. c 4. b 5. b

SUBIECTUL al II-lea

1. Rezolvare:

a) 9553

b) Algoritmul adună dintre cele n numerele citite pe cele care respectă condiția: numărul obținut prin eliminarea cifrelor strict mai mari decât c este par. Un set de numere posibile: 5, 6, 34, 128, 12, 26, 179, deoarece se va obține suma egală cu $34+128+12+26=200$

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    unsigned int n,c,e,i,x,p,nr,cx;
    cin>>n>>c;
    e=0;
    for(i=1;i<=n;i++){
        cin>>x;
        p=1;
        nr=0;
        cx=x;
        while(x!=0){
            if(x%10<=c){
                nr=nr+(x%10)*p;
                p=p*10;
            }
            x=x/10;
        }
        if(nr%2==0 && nr>0)
            e=e+cx;
    }
    cout<<e;
    return 0;
}
```

d) Programul în limbaj pseudocod echivalent este :

```
citește n,c (n, c numere naturale nenule, c<10)
e←0
i←1
cât timp i≤n execută
    p←1
    nr←0
    cx←x
    cât timp x≠0 execută
        dacă x%10≤c atunci
            nr←nr+(x%10)*p
            p←p*10
        x←[x/10]
    dacă nr%2=0 și nr>0 atunci
        e←e+cx
    i←i+1
scrie e
```

```
2. cin.getline(p,50);
for(i=0;i<200;i++){
    if(strcmp(p,c[i].proba)==0 && c[i].punctaj==100){
        cout<<c[i].nume_concurent<<',';
    }
}
```

3. $a[i][j]=(i+j)\%5;$

SUBIECTUL al III-lea

1.

```

void perechi(int n, int v[], int &p){
    int a,b,r,i,j;
    p=0;
    for(i=1;i<n;i++){
        for(j=i+1;j<=n;j++){
            a=v[i];
            b=v[j];
            while(b!=0){
                r=a%b;
                a=b;
                b=r;
            }
            if(a==1)
                p++;
        }
    }
}

```

2.

```

#include<iostream>
#include<cstring>
using namespace std;

int main()
{
    char c[21], s[201],*p;
    int n,nr,i,j;
    cin>>n>>c;
    cin.get();
    for(i=1; i<=n; i++)
    {
        cin.getline(s,201);
        for(j=0; j<strlen(s); j++)

            if(s[j]>='A' && s[j]<='Z')
            {
                s[j]=s[j]+32;
            }
        nr=0;
        p=strtok(s," ");
        while(p!=NULL)
        {
            if(strstr(p,c)==p)
                nr++;
            p=strtok(NULL," ");
        }
        cout<<nr<<' ';
    }
    return 0;
}

```

3. a. Programul utilizează un vector v cu 1000 de elemente, corespunzător valorilor posibile ale numerelor citite, în care se completează valoarea 1 pentru numerele din șirul a simultan cu citirea lor. La citirea fiecărui număr x din șirul b se verifică dacă $v[x]=0$ (ce semnifică faptul că nu există valoarea x în șirul a și poate fi inserată) și se completează $v[x]$ cu valoarea 2. Algoritmul este eficient ca spațiu de memorie deoarece utilizează un singur vector și este eficient ca timp de execuție deoarece parcurge o singură dată fiecare șir de numere (algoritm liniar).

b.

```
#include<iostream>
#include<fstream>
using namespace std;
int v[1000];
int main(){
    ifstream fin("BAC.TXT");
    int n,m,i,a,b,ok;
    ok=0;
    fin>>n>>m;
    for(i=1;i<=n;i++){
        fin>>a;
        v[a]=1;
    }
    for(i=1;i<=m;i++){
        fin>>b;
        if(v[b]==0){
            v[b]=2;
            ok=1;
        }
    }
    if(ok==0)
        cout<<"Nu exista numere care vor fi inserate";
    else{
        for(i=0;i<=999;i++)
            if(v[i]==2)
                cout<<i<<' ';
    }
    return 0;
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 2**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

2. d	2. c	3. c	4. b	5. b	5x4p.
-------------	-------------	-------------	-------------	-------------	--------------

SUBIECTUL al II - lea**(40 de puncte)**

	a) Răspuns corect: 9553	6p.	
	b) Pentru răspuns corect	6p.	
1.	c) Pentru program corect -declaraire variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă numai una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -citire corectă a denumirii probei p -acces la câmpurile înregistrării -afișarea rezultatelor cerute (*) -corectitudine globală a secvenței de program	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (determinarea numelor concurenților, determinarea probei p, determinarea punctajului egal cu 100,)
3.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*)	6p. 1p. 5p.	(*) Se acordă numai 2p. dacă doar o parte dintre elementele tabloului sunt conform cerinței.

¹ Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.

SUBIECTUL al III - lea**(30 de puncte)**

1.	Pentru subprogram corect -antet subprogram (*) -determinare a numărului cerut (**) -returnarea rezultatului prin parametru -declarare a variabilelor locale, -corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	.(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri) conform cerinței. (**) Se acordă câte 3p. pentru fiecare aspect al cerinței (algoritm de verificare dacă două numere sunt prime între ele, algoritm de determinare a valorii p).
2.	Pentru program corect -declararea variabilelor -citire a datelor -determinare a șirului de valori cerut (*) -declarare a variabilelor simple, -corectitudine a globală a programului ¹⁾	10p. 1p. 2p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței (acces la o linie din text, cuvinte suport, identificare prefix, determinare număr de cuvinte care sunt prefixe, afișarea numărului de prefixe din linie, utilizarea tuturor liniilor din text).
3.	a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență	2p. 1p. 1p.	.(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este
	b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizarea unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	8p. 1p. 5p. 1p. 1p.	principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă construiește, pe măsura citirii datelor, un vector caracteristic pentru elementele din șirul a și le notează cu 1. Apoi citește elementele șirului b și verifică pentru fiecare număr, pe baza vectorului caracteristic, dacă nu există un termen egal cu el din șirul a , în caz afirmativ acesta se notează cu 2 în vector. Se parcurge vectorul caracteristic, în sens crescător și se afișează numerele notate cu 2.

MODEL TEST 3

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Care dintre următoarele expresii are valoarea 1, dacă **a** este o variabilă întreagă cu valoarea 2020?

- a. $(a \% 3 / 10 == a / 10 \% 3) \&\& (a \% 3)$ b. $(a \% 2 / 10 == a / 100 \% 2) \&\& (a \% 11)$
- c. $(a == a - 1) \&\& (a \% 7)$ d. $(a / 20 != 101) \&\& (a \% 5)$

2. Ce valoare va avea în urma apelului **f(2020)** variabila globală **a** de tip întreg, dacă înainte de apel **a=0** și funcția **f** este definită alături?

```
int f(int n)
{
    a++;
    if(!n) return 1;
    return f(n/100)+f(n/10);
}
```

- a. 1 b. 0 c. 15 d. 8

3. Într-o sală de cinema sunt prezentate patru filme numerotate cu 1, 2, 3 și 4. Utilizând metoda Backtracking, se generează toate posibilitățile de a le prezenta pe toate știind că filmul 2 se difuzează după filmul 3, într-o ordine nu neapărat consecutivă. Câte modalități de prezentare există?

- a. 12 b. 16 c. 8 d. 6

4. Un arbore cu rădăcină are 7 noduri numerotate de la 1 la 7. Nodurile de grad >1, **i** sunt adiacente cu nodurile **2*i** și **2*i+1**. Știind că nodul 7 este nod rădăcină, care dintre următoarele variante reprezintă vectorul de tați?

- a. **T=(2,7,1,3,3,2,0)** b. **T=(2,1,7,3,3,2,0)**
- c. **T=(3,7,7,2,2,3,0)** d. **T=(3,1,7,2,2,3,0)**

5. Pentru graful neorientat cu 105 de noduri în care toate nodurile au același grad, care dintre următoarele variante poate reprezenta gradul unui nod?

- a. 13 b. 1 c. 3 d. 6

SUBIECTUL al II-lea**(40 de puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[x]$ partea întregă a lui x .

- a. Dacă pentru a se citește valoarea 14, iar pentru b valoarea 93 ce va afișa algoritmul? **(4 p)**
- b. Dacă pentru a se citește valoarea 5, iar algoritmul afișează $172/33$, atunci ce valoare trebuie să aibă b ? **(6 p)**
- c. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structurile cât timp...execută cu structuri de alt tip. **(4 p)**
- d. Scrieți programul C/C++ corespunzător algoritmului dat. **(6 p)**

```

citește a,b (b≠0)
f2←0
x←b
cât timp x≠0 execută
  f2←f2*10+9
  x←[x/10]
  ■
f1←a*f2+b
a←f1
b←f2
cât timp b≠0 execută
  r←a%b
  a←b
  b←r
  ■
scrie f1/a,',f2/a

```

2. O structură de date neomogenă memorează date despre elevii unei clase: numele are cel mult 30 de caractere, două note de tip întreg și media notelor. Scrieți definiția completă a structurii și declarați corespunzător variabila x , astfel încât să se poată realiza următoarea operație: $x.media=9.5$; **(10 p)**

3. Pentru o matrice a cu 7 linii și 7 coloane și fără a folosi alte variabile, stabiliți cu ce trebuie înlocuite punctele de suspensie pentru ca după executarea instrucțiunilor să se obțină matricea alăturată:

```

for(i=0;i<7;i++)
  for(j=0;j<7;j++)
    ...

```

```

31 33 35 37 39 41 43
28 30 32 34 36 38 40
25 27 29 31 33 35 37
22 24 26 28 30 32 34
19 21 23 25 27 29 31
16 18 20 22 24 26 28
13 15 17 19 21 23 25

```

(10 p)**SUBIECTUL al III-lea****(30 de puncte)**

1. Se citește un șir de maxim 200 de caractere. Scrieți un program care să construiască în memorie și să afișeze un șir de caractere ce conține toate caracterele din șirul citit și în care se înserează după fiecare vocală sau litera 'c', șirul "STOP".

De exemplu, dacă șirul este "aCc, cdE", atunci se obține șirul "aSTOPCcSTOP, cSTOPdESTOP". **(10 p)**

2. Să se scrie definiția unei funcții numită **duplicat** care primește ca parametri vectorul a și valoarea n ($<10^4$). Vectorul conține toate valorile de la 1 la $n-1$, astfel că o singură valoare se repetă. Funcția va returna valoarea care se repetă. Elementele vectorului sunt indexate de la 0.

De exemplu, dacă $a=(4,2,3,1,2)$ și $n=5$, atunci funcția va returna valoarea 2. **(10 p)**

3. Fișierul **info.txt** conține pe prima linie o valoare naturală n ($<10^5$), iar pe următoarea linie un șir de n numere naturale ($<10^9$) separate prin spații.

a. Scrieți un program care să citească din fișier șirul de numere și care determină eficient din punct de vedere al timpului de executare și al memoriei, câte dintre perechile de elemente din șir sunt formate din valori cu aceeași sumă a cifrelor. **(8 p)**

b. Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei. **(2 p)**

De exemplu, pentru $n=4$ și numerele 24, 12, 21, 33 rezultatul este 2.

REZOLVARE SUBIECTE
TEST 3
SUBIECTUL I

1. b
2. c
3. a
4. d
5. d

SUBIECTUL II

1.

a. 493/33

b. 21

c. citește a,b (b≠0)

f2←0

x←b

repetă

f2←f2*10+9

x←[x/10]

■ până când x=0

f1←a*f2+b

a←f1

b←f2

repetă

r←a%b

a←b

b←r

■ până când b=0

scrie f1/a,','f2/a

d.

```
#include<iostream>
using namespace std;
int a,b,f1,f2,r,x;
int main()
{
    cin>>a>>b;
    x=b;
    while(x)
        f2=f2*10+9, x=x/10;
    f1=a*f2+b;
    a=f1;
    b=f2;
    while(b)
        r=a%b, a=b, b=r;
    cout<<f1/a<<' '<<f2/a;
}
```

2.

struct elev

{

char nume[31];

int nota1, nota2;

double media;

};

3.

a[i][j]=10+3*(7-i)+2*j;

SUBIECTUL III

1.

#include<iostream>

#include<cstring>

using namespace std;

char s[201],t[1001];

int i;

int main()

{

cin.getline(s,201);

for(i=0;s[i];i++)

{

t[strlen(t)]=s[i];

if(strchr("aceiouAEIOU",s[i]))

strcat(t,"STOP");

}

cout<<t;

}

2.

```
int duplicat(int a[],int n)
{
    int s=0,i;
    for(i=0;i<n;i++)
        s=s+a[i];
    return s-n*(n-1)/2;
}
```

3.

a.

```
#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("info.txt");
int v[82],n,i,k,s;
int main()
{
    fin>>n;
    while(n-->0)
    {
        fin>>i;
        s=0;
        while(i)
            s=s+i%10,i=i/10;
        v[s]++;
    }
    for(i=0;i<82;i++)
        k=k+v[i]*(v[i]-1)/2;
    cout<<k;
}
```

b.

Se utilizează un vector \mathbf{v} în care $\mathbf{v}[\mathbf{i}] =$ câte numere citite au suma cifrelor \mathbf{i} . Algoritmul este eficient din punct de vedere al timpului de executare deoarece se parcurg o singură dată numerele din fișier și este eficient din punct de vedere al memoriei deoarece se folosesc variabile simple și un vector de 82 de elemente.

BAREM DE EVALUARE ȘI DE NOTARE

TEST 3

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.

SUBIECTUL I

20 de puncte

1.	b	4p	
2.	c	4p	
3.	a	4p	
4.	d	4p	
5.	d	4p	

SUBIECTUL II

40 de puncte

1.	a	493/33	4p	
	b	21	6p	
	c	Pentru rezolvare corectă	4p.	Se acordă numai 2 puncte dacă doar una dintre cele două instrucțiuni este corectă
	d	Pentru program corect -declararea corectă a tuturor variabilelor -citire corectă -scriere corectă -instrucțiune repetitivă corectă -atribuiri corecte -corectitudinea globală a programului.	6p. 1p. 1p. 1p.. 1p. 1p. 1p	
2.		Pentru rezolvare corectă -denumire corectă a structurii -declarare corectă a șirului de caractere -declarare corectă a câmpurilor pentru note -declarare corectă a câmpului pentru medie -declararea variabilei x -corectitudine globală	10p 1p 2p 2p 2p 2p 1p	
3.		Pentru rezolvare corectă	10p.	

SUBIECTUL III

30 de puncte

1.	Pentru program corect -citirea datelor -afișarea datelor -identificarea literelor după care se face inserarea -modificarea șirului(*) -corectitudine globală	10p. 1p 1p 2p 5p 1p	(*)Se acordă numai 2p. dacă algoritmul este principal corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare.
2.	Pentru subprogram corect -antet corect -parametri corespunzători -identificarea dublurii(*) -returnarea valorii cerute -corectitudine globală	10p. 1p 1p 5p 2p 1p	(*)Se acordă numai 2p. dacă algoritmul este principal corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare.

3.	a. Pentru program corect --operații cu fișiere: declarare, citire/ scriere din/în fișier -utilizare a unui algoritm eficient (**) -determinare și afișare a rezultatului conform cerinței,(***) -declarare variabile, corectitudine globală a programului	8p. 1p 2p 4p 1p	(**) Se acordă punctajul numai pentru un algoritm (de complexitate $O(n)$), care utilizează eficient memoria. (***) Se acordă numai 2p. dacă algoritmul este principial corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare.
	b. Pentru răspuns corect -coerența explicării metodei (*) -justificare a unor elemente de eficiență	2p. 1p 1p	(*) Se acordă punctajul chiar dacă metoda aleasă nu este eficientă.

Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.

- a. Scrieți valoarea afișată dacă se citesc valorile 51467 și 67. **(6p)**
- b. Dacă pentru **b** se citește valoarea 73, scrieți numărul valorilor de 4 cifre ce pot fi citite pentru a astfel încât valoarea afișată să fie divizibilă cu 50. **(6p)**
- c. Scrieți un algoritm echivalent cu cel dat care să utilizeze o structură repetitivă cu test final. **(6p)**
- d. Scrieți programul C/C++ corespunzător algoritmului dat **(10p)**

2. În secvența de instrucțiuni de mai jos, variabilele *i* și *j* sunt de tip întreg, iar variabila **A** este de tip tablou bidimensional cu 5 linii și 5 coloane cu numere întregi. Fără a utiliza alte variabile, înlocuiți cu o instrucțiune punctele de suspensie din secvența de mai jos astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă conținutul alăturat:

```
for (i=1;i<=5;i++)
    for (j=1;j<=5;j++)
        .....
```

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 0 \\ 2 & 4 & 1 & 3 & 0 \\ 3 & 1 & 4 & 2 & 0 \\ 4 & 3 & 2 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$
(6p)

3. În declarația alăturată, variabila **A** memorează în câmpurile **x** și **y** coordonatele carteziene ale unui punct din planul **xOy**, iar variabila **C** memorează raza și coordonatele centrului unui cerc.

```
struct punct
    {float x,y;} A;
struct cerc
    {punct p; float r;} C;
float d;
```

Scrieți o secvență de instrucțiuni prin care se calculează în variabila **d** distanța de la punctul **A** la centrul cercului și apoi se verifică poziția punctului față de cerc afișându-se, în funcție de rezultatul obținut, mesajul INTERIOR, EXTERIOR, respectiv PE CERC.

(6p)

Subiectul III

(30 de puncte)

1. Subprogramul **nrmax** primește prin parametrii **a** și **b** două numere naturale cu cel mult 9 cifre fiecare și returnează prin parametrul **c** cel mai mare număr cu cifrele impare distincte care apar atât în **a** cât și în **b** sau 0 dacă nu există astfel de cifre.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **a=45013** și **b=215578** atunci se obține **c=51**.

(10 p)

2. Se consideră un text cu maxim 250 de caractere (litere mici și spații) în care cuvintele sunt separate printr-un singur spațiu. Scrieți un program care citește un astfel de text și apoi construiește șirul obținut prin modificarea tuturor cuvintelor de lungime pară prin interschimbarea între ele a celor două jumătăți ale cuvântului, păstrând ordinea caracterelor din cele două jumătăți. Programul afișează pe ecran textul astfel modificat, iar în cazul în care nu sunt cuvinte modificate se va afișa mesajul **TEXT NEMODIFICAT**.

Exemple: pentru textul **totul va fi bine** se va obține textul **totul av if nebi**, iar pentru textul **mama tata** se va afișa mesajul **TEXT NEMODIFICAT**

(10 p)

3. Fișierul **BAC.IN** conține cel mult un milion de numere naturale de cel mult 9 cifre fiecare separate prin câte un spațiu. Să se afișeze pe ecran cel mai mare număr **a** de exact 3 cifre care nu apare în fișier pentru care există un număr **b**, care apare în fișier, astfel încât **a+b=1000**.

Dacă nu există niciun astfel de număr se va afișa mesajul **NU EXISTA**.

Proiectați un algoritm eficient din punct de vedere al timpului de executare și justificați eficiența acestuia.

Exemple:

BAC.IN	22 8425 4567 6 8 999 33 995 800 200 45123	8425 994 4567 5 6 994 3300 995 800 200 45123
Pe ecran	994	NU EXISTA

(10 p)

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I

1. b	2. d	3. b	4. d	5. d
------	------	------	------	------

Subiectul II

1. a. 51400

b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$, $x \in \{1, 2, \dots, 9\}$, $y \in \{0, 1, \dots, 9\}$)

c. citește a, b (numere naturale)
 $p \leftarrow 1$
 dacă $a*b > 0$ și $a \% 10 = b \% 10$ atunci
 repetă
 $a \leftarrow [a/10]$
 $b \leftarrow [b/10]$
 $p \leftarrow p*10$
 până când $a*b = 0$ sau $a \% 10 \neq b \% 10$
 $a \leftarrow a*p$
 scrie a

d.

```
#include<iostream>
using namespace std;
int main()
{ int a,b,p=1;
cin>>a>>b;
while (a*b>0 && a%10==b%10)
{ a=a/10;
  b=b/10;
  p=p*10;
}
a=a*p;
cout<<a;
return 0;
}
```

2. $A[i][j] = (i*j) \% 5$;

```
3. d=sqrt((A.x-C.p.x)*(A.x-C.p.x)+(A.y-C.p.y)*(A.y-C.p.y));
if (d<C.r) cout<<"INTERIOR";
else if (d>C.r) cout<<"EXTERIOR";
else cout<<"PE CERC";
```

Subiectul III

```
1. void nrmax (long a, long b, long &c)
{ int i, ok1, ok2; long a1, b1;
c=0;
for (i=9;i>=1;i-=2)
{ ok1=ok2=0; a1=a; b1=b;
while (a1>0&&ok1==0) if (a1%10==i)ok1=1;
else a1/=10;
while (b1>0&&ok2==0) if (b1%10==i) ok2=1;
else b1/=10;
if (ok1==1 && ok2==1) c=c*10+i;
}
}
```

```

2. #include<iostream>
   #include<cstring>
   using namespace std;
   int main()
   { char s[251], t[ ], c[ ], *p, z[ 251];
     int ok=0;
     cin.get (s, 251);
     z[0]=NULL;
     p=strtok(s," ");
     while (p) { if (strlen(p)%2==0){
                   strcpy(c,p);
                   strcpy(t, p+strlen(p)/2);
                   p[strlen(p)/2]=NULL;
                   strcat (t,p);
                   if (strcmp (c,t)) {
                       ok=1;
                       strcpy(p,t);
                   }
                   }
                   strcat(z,p);
                   strcat(z," ");
                   p=strtok(NULL," ");
           }
     if (ok) cout <<z;
     else cout << "TEXT NEMODIFICAT";
     return 0;
   }
3. #include<iostream>
   #include<fstream>
   using namespace std;
   int v[1000];
   ifstream f ("BAC.IN");
   int main()
   {long x, ok=0;
     while (f>>x) if (x<1000)v[x]++;
     x=999;
     while (x>=100&&ok==0)
         if (v[x]==0 && v[1000-x]!=0) ok=1;
         else x--;
     if (ok) cout<<x;
     else cout<<"NU EXISTA";
     return 0;
   }

```

O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifica primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Complexitatea algoritmului este liniară ($O(n)$).

BAREM DE EVALUARE ȘI DE NOTARE**TEST 4**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I**20 de puncte**

1. b	2. d	3. b	4. d	5. d	4p×5
------	------	------	------	------	------

Subiectul II**40 de puncte**

1. a. 51400	6p
b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$)	6p
c. echivalare corectă	6p
d. Pentru program corect	10p
-declararea corectă a tuturor variabilelor	2p
-citire corectă	1p
-scriere corectă	1p
-instrucțiune repetitivă corectă	3p
-atribuiri corecte	2p
-corectitudinea globală a programului	1p
2. $A[i][j]=(i*j)\%5;$	6p
3. Pentru rezolvare corectă	6p
-accesul corect la câmpurile de pe primul nivel al înregistrării	1p
-accesul corect la câmpurile de pe al doilea nivel al înregistrării	1p
-determinarea corectă a valorii variabilei d	2p
-instrucțiunea de decizie care afișează corect mesajul cerut	2p

Subiectul III**30 de puncte**

1. Pentru subprogram corect	10 p
-structură antet corect	2p
-declarare corectă a parametrilor	1p
-determinare cifre distincte comune	3p
-determinare număr maxim cerut	3p
-declarare a tuturor variabilelor locale, corectitudine globală a subprogramului	1p
2. Pentru rezolvare corectă	10p
-declarare corectă a unei variabile care să memoreze un șir de caractere	1p
-citirea șirului de caractere	1p
-separarea cuvintelor din frază	2p
-modificarea corectă a cuvintelor cu lungimea pară	2p
-construirea corectă a șirului cerut	2p
-afișarea mesajului	1p
-corectitudine globală a programului	1p
3. Pentru rezolvare corectă	10p
- operații cu fișiere (declarare, deschidere)	1p
- citirea numerelor	1p
- algoritm principal corect	2p
- determinarea valorii cerute	2p
- afișarea rezultatului	1p
- corectitudine formală (declarare variabile, structură program, sintaxa instrucțiunilor)	1p
- coerența explicării metodei	1p

<p>- explicarea unor elemente de eficiență conform cerinței (numai pentru eficiență $O(n)$)</p> <p>O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifică primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Astfel, algoritmul utilizat este de tip liniar (complexitate $O(n)$).</p>	1p
---	----

MODEL TEST 5

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I
(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns se notează cu 4 puncte.

- Variabila **n** memorează un număr natural cu 5 cifre. Care dintre expresiile C/C++ de mai jos schimbă cifra din mijloc cu cifra 0?

a) $n=n\%1000+n/100*100$	c) $n=n\%100+n/1000*1000$
b) $n=n\%10+n/10*10+n/100*100$	d) $n=n/10\%100*1000+n/100$
- Fiind dat un graf complet, se știe că pentru a obține 3 componente conexe, trebuie să eliminăm minim 9 muchii. Care este numărul minim de noduri pe care ar trebui să îl aibă graful inițial?

a) 8	c) 7
b) 9	d) 6
- Utilizând metoda backtracking, se generează toate șirurile de câte patru operatori din mulțimea $\{ '+', '-', '*', '/', '%' \}$, șiruri în care nu pot fi alăturate primul și ultimul operator. Primele 8 șiruri sunt: $+++-$, $+++*$, $+++/$, $++++$, $+++*$, $++-/$, $+++%$. Câte dintre aceste șiruri generate încep cu '-' și se termină cu '%'?

a) 13	c) 10
b) 15	d) 20
- Se consideră un arbore cu 9 noduri, numerotate de la 1 la 9, cu rădăcina nodul 1 în care avem muchiile: $[1,2]$, $[1,3]$, $[1,4]$, $[4,5]$, $[4,6]$, $[5,7]$, $[5,8]$, $[7,9]$. Care sunt ascendenții nodului 7?

a) 5, 6	c) 1, 4, 5
b) 1, 2, 3, 4, 5, 6	d) 1, 4, 6
- Se consideră subprogramul **f** definit alăturat. În urma cărui apel valoarea returnată de subprogram va fi **8**?

<table border="0"> <tr> <td>a) $f(34)$</td> </tr> <tr> <td>b) $f(10)$</td> </tr> <tr> <td>c) $f(128)$</td> </tr> <tr> <td>d) $f(256)$</td> </tr> </table>	a) $f(34)$	b) $f(10)$	c) $f(128)$	d) $f(256)$	<pre> int f(int n){ if(n==1) return 0; if(n%2==0) return 1+f(n/2); else return 1+f(3*n+1); } </pre>
a) $f(34)$					
b) $f(10)$					
c) $f(128)$					
d) $f(256)$					

SUBIECTUL al II-lea**(40 de puncte)****1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu $a\%b$ restul împărțirii numărului natural a la numărul natural b .

- a) Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele **4, 3, 12, 36, 45, 51, 27, 87, 17, 25**. **(6p.)**
- b) Dacă pentru n se citește 2, iar pentru p se citește 5, scrieți patru numere distincte care pot fi citite, astfel încât în urma executării algoritmului, valoarea afișată să fie **0**. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască structura **repetă...până când** cu o structură de tip **pentru...execută**. **(6p.)**

```

citește n, p
(numere naturale nenule)
nr ← 0
repetă
    citește x, y
    (numere naturale)
    -- cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    ■
    dacă (x = p) atunci
        nr ← nr + 1
    ■
    până când n = 0
scrie nr

```

2. Fie următoarea definiție și declarație:

```

struct examen{
    char nume[30], prenume[30], initiala;
    float nota[3], medie;
};e[300];

```

Să se scrie secvența de program care citește pentru n candidați la examenul de bacalaureat următoarele date: numele, inițiala tatălui, prenumele, notele la cele trei probe și calculează media obținută la examen. În cazul în care candidatul a promovat cele trei probe și examenul de bacalaureat să se afișeze media sau mesajul **respins** în caz contrar (un candidat este promovat dacă are cel puțin nota 5 la fiecare probă și media cel puțin 6).

(6p.)

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg, iar variabila A memorează un tablou bidimensional cu 4 linii și 6 coloane, numerotate începând de la 1. Elementele tabloului sunt numere întregi. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila A să aibă elementele din figura de mai jos. **(6p.)**

```

for(i=1;i<=4;i++)
    for (j=1;j<=6;j++)
        .....

```

0	1	1	1	1	1
1	0	1	0	1	0
0	1	0	3	3	3
1	2	3	0	1	2

SUBIECTUL al III-lea**(30 de puncte)**

1. Un număr natural n , având un număr de k cifre, se numește număr **Armstrong** dacă este egal cu suma cifrelor sale ridicate la puterea k .

Subprogramul **armstrong** are un singur parametru, n , prin care primește un număr natural ($n \in [0, 10^9]$). Subprogramul returnează **1** dacă n este număr Armstrong și **0** în caz contrar. Scrieți definiția completă a subprogramului. **(10p.)**

Exemplu: dacă $n=153$ subprogramul returnează 1 ($153=1^3+5^3+3^3$).

2. Într-un text cu cel mult **100** de caractere, cuvintele sunt formate din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un text de tipul menționat și afișează pe ecran, pe linii separate, toate cuvintele care încep și se termină cu o consoană, iar în rest nu conțin decât vocale. Dacă nu există niciun astfel de cuvânt, se afișează pe ecran mesajul **nu exista**. Se consideră vocale literele din mulțimea $\{a, e, i, o, u\}$. **(10p.)**

Exemplu: pentru textul **fetita a desenat pe caiet un deal acoperit cu verdeata si un cal cu coama neagra** se afișează pe ecran, nu neapărat în această ordine, cuvintele de mai jos:

caiet

deal

cal

3. Numerele întregi pozitive cu proprietatea că, prin însumarea iterativă a pătratelor cifrelor lor, se ajunge în cele din urmă la numărul 1, se numesc **numere fericite**. Numărul 7 este un număr fericit pentru că $7^2=49$, $4^2 + 9^2=97$, $9^2 + 7^2=130$, $1^2 + 3^2 + 0^2=10$, $1^2 + 0^2=1$. Prelucrând astfel orice număr, în cele din urmă se va ajunge doar la unul dintre următoarele numere posibile: 0, 1, 4, 16, 20, 37, 42, 58, 89 sau 145.

Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze în fișierul **bac.out**, pe câte un rând, numerele din șir care sunt *fericite* urmate de numărul de iterații necesare pentru a ajunge la numărul 1. Dacă în șir nu există niciun număr fericit se va afișa mesajul **nu exista**. Proiectați un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate.

Exemplu:

bac.in	bac.out
7 13 95 104 86 17 379 226 445 33	7 5
	13 2
	86 2
	379 6
	226 5

- a) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
 b) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

Subiectul I

1. c
2. d
3. b
4. c
5. d

Subiectul II

1. Rezolvare:

a) 2

n	p	nr	x	y	z
4	3	0	12	36	12
			36	12	0
			12	0	
3		1	45	51	45
			51	45	6
			45	6	3
			6	3	0
			3	0	
2		2	27	87	27
			87	27	6
			27	6	3
			6	3	0
			3	0	
1			17	25	17
			25	17	8
			17	8	1
			8	1	0
			1	0	
0					

b) Algoritmul numără câte seturi de numere x, y au cel mai mare divizor comun numărul p .

Un set de numere posibile: 64, 26, 13, 29

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    int n,p,nr,x,y,z;
    cin>>n>>p;
    nr=0;
    do{
        cin>>x>>y;
        while(y!=0){
            z=x%y;
            x=y;
            y=z;
        }
        if(x==p)
            nr++;
        n--;
    }while(n!=0);
    cout<<nr;
    return 0;
}
```

d) Pseudocod echivalent:

```
citește n, p (numere naturale)
nr ← 0
pentru i ← 1, n, 1 execută
    citește x, y (numere naturale)
    cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    dacă (x = p) atunci
        nr ← nr + 1
scrie nr
```

2.

```
int i,n;
for(i=1;i<=n;i++){
    cin.getline(e[i].nume,30);
    cin.getline(e[i].prenume,30);
    cin>>e[i].initiala>>e[i].nota[0]>>e[i].nota[1]>>e[i].nota[2];
    if(e[i].nota[0]>=5 && e[i].nota[1]>=5 && e[i].nota[2]>=5){
        e[i].medie=(e[i].nota[0]+e[i].nota[1]+e[i].nota[2])/3;
        if(e[i].medie>=6)
            cout<<e[i].medie;
        else
            cout<<"respins";
    }
    else cout<<"respins";
}
```
3.

```
if(i%2==1)
    a[i][j]=i*j;
else a[i][j]=j*i;
```

Subiectul III

1. Rezolvare:

```
int armstrong(int n){
    int k, aux,s,i,p;
    k=s=0;
    aux=n;
    while(aux!=0){
        k++;
        aux=aux/10;
    }
    aux=n;
    while(aux!=0){
        p=1;
        for(i=1;i<=k;i++){
            p=p*(aux%10);
        }
        s=s+p;
        aux=aux/10;
    }
    if(n==s)
        return 1;
    else
        return 0;
}
```

2. Rezolvare:

```
#include <iostream>
#include<string.h>
using namespace std;
int main(){
    char s[100], *p;
    int i,ok,n,k=0;
    cin.getline(s,100);
    p=strtok(s," ");
    while(p){
        n=strlen(p);
        if(strchr("aeiou",p[0])==NULL && strchr("aeiou",p[n-1])==NULL){
            ok=1;
            for(i=1;i<strlen(p)-1;i++){
                if(strchr("aeiou",p[i])==NULL)
                    ok=0;
            }
            if(ok==1){
                cout<<p<<endl;
                k++;
            }
        }
        p=strtok(NULL," ");
    }
    if(k==0)
        cout<<"nu exista";
    return 0;
}
```


3. Rezolvare:

```
#include <iostream>
#include<fstream>
using namespace std;
int main(){
    ofstream fout("bac.out");
    ifstream fin ("bac.in");
    int s,x,it,aux,ok;
    ok=0;
    while(fin>>x){
        aux=x; it=0;
        do{
            s=0; it++;
            while(x!=0){
                s=s+x%10*(x%10);
                x=x/10;
            }
            x=s;
        }while(s!=1 && s!=0 && s!=4 && s!=16 && s!=20 && s!=37 && s!=42
&& s!=58 && s!=89 && s!=145);
        if(s==1){
            fout<<aux<<' '<<it<<endl;
            ok=1;
        }
    }
    if(ok==0)
        fout<<"nu exista";
    return 0;
}
```

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului

SUBIECTUL I**(20 de puncte)**

1c 2b 3b 4c 5d	5 x 4p.
-----------------------	----------------

SUBIECTUL al II-lea**(40 de puncte)**

1.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare set de numere x, y conform cerinței (un set de numere x, y este corect dacă c.m.m.d.c (x,y) este diferit de 5).
	c) Pentru program corect -declorare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -acces la câmpurile înregistrării -citire date -verificare a condiției impuse (*) -corectitudine globală a expresiei ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței referitor la condiția impusă (note, medie, operatori logici utilizați conform cerinței).
3.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (atribuire valori, identificare linii cu indice par/linii cu indice impar). O soluție posibilă este atribuirea valorii expresiei i%j elementelor aflate pe linii cu indice impar, respectiv j%i elementelor aflate pe linii cu indice par.

SUBIECTUL al III-lea**(30 de puncte)**

1.	Pentru subprogram corect -antet subprogram (*) -verificare a proprietății cerute (**) -instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, declarare parametru de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unei cifre, ridicarea unui număr la o putere, calcul sumă).
2.	Pentru program corect -declarare a unei variabile care să memoreze un șir de caractere -citire a datelor -determinare a cuvintelor cerute (*) -afișare a datelor în formatul cerut și tratare a cazului nu exista -declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (obținere a unui cuvânt, localizare a consoanelor pe prima și ultima poziție a unui cuvânt, localizare a vocalelor).
3.	a) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii/scrierii, citire/scriere din/în fișier -determinare a valorilor cerute (*), (**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului b) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență	8p. 1p. 5p. 1p. 1p. 2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă parcurge șirul din fișier, memorând valoarea curentă, calculează iterativ suma pătratelor cifrelor cât timp nu s-a ajuns la o valoare particulară și numără iterațiile efectuate.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

MODEL TEST 6

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **x** memorează un număr natural de exact 5 cifre. Care dintre următoarele expresii are ca valoare un număr natural format din cele 3 cifre situate în mijlocul numărului memorat în variabila **x**?

- a. $x \% 10$
- b. $x / 10$
- c. $x / 10 \% 1000$
- d. $x \% 1000 / 10$

2. Se consideră **z** o variabilă globală care are valoarea 7 și următorul subprogram **f**:

```
int z=7;  
void f(int x, int &y)  
{ x++; y+=x; z+=y; }
```

Care va fi valoarea variabilei **z** după apelul **f(z,z)**?

- a. 23
- b. 32
- c. 7
- d. 30

3. Să considerăm două variabile **P1** și **P2**, care rețin coordonatele carteziene a două puncte în plan, declarate astfel:

```
struct Punct {double x, y;} P1, P2;
```

Care dintre următoarele variante reprezintă lungimea segmentului determinat de punctele memorate în variabilele **P1** și **P2**?

- a. $\text{sqrt}((\text{P1.x-P1.y}) * (\text{P1.x-P1.y}) + (\text{P2.x-P2.y}) * (\text{P2.x-P2.y}))$;
- b. $\text{sqrt}((\text{P1.x-P2.x}) * (\text{P1.x-P2.x}) + (\text{P1.y-P2.y}) * (\text{P1.y-P2.y}))$;
- c. $\text{sqr}((\text{P1.x-P2.x}) * (\text{P1.x-P2.x}) + (\text{P1.y-P2.y}) * (\text{P1.y-P2.y}))$;
- d. $\text{sqrt}((\text{P1.x-P2.x})^2 + (\text{P1.y-P2.y})^2)$;

4. Utilizând metoda backtracking, se generează toate secvențele formate din **n** caractere din mulțimea {'a', 'b', 'o'}, astfel încât în orice prefix al unei secvențe astfel generate numărul de caractere 'b' nu depășește numărul de caractere 'a'. De exemplu, pentru **n=3**, s-au generat (în această ordine) următoarele secvențe:

aaa, aoa, aab, aoa, aoo, aob, aba, abo, oaa, oao, oab, ooa, ooo.

Dacă utilizăm același algoritm pentru a genera secvențele de lungime **n=4**, care va fi cea de a opta secvență generată?

- a. abbo
- b. aabb
- c. aabo
- d. aaba

5. Un graf neorientat are **100** de vârfuri și **7** componente conexe. Numărul maxim de muchii pe care le poate avea acest graf este:
- a. 3822 b. 4950 c. 8742 d. 4371

SUBIECTUL al II-lea**(40 de puncte)****1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întreagă a numărului natural x .

- a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește numărul **29357**. (6p.)
- b. Scrieți o valoare de **5** cifre care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze **123**. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină alt tip de structuri repetitive. (6p.)

e.
citește n (număr natural)
 $x \leftarrow 0$
repetă
 $x \leftarrow x * 10 + n \% 10$
 $n \leftarrow [n / 10]$
până când $n = 0$
repetă
 $n \leftarrow n * 10 + x \% 10$
 $x \leftarrow [x / 100]$
până când $x = 0$
scrie n

2. În declarația alăturată, variabila x memorează, pentru fiecare dintre cei 30 de elevi dintr-o clasă, numărul matricol, precum și data nașterii elevului. Scrieți o secvență de instrucțiuni C/C++ care citește de la tastatură informațiile despre primul elev memorat și afișează pe ecran numărul matricol al elevului, dacă anul nașterii este mai mic decât 2013, sau mesajul **imposibil** în caz contrar. (6p.)

```
struct elev
{ int nr_mat;
  struct
  { int zi,luna,an;
  }data;
}x[30];
```

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional de numere întregi cu **7** linii și **7** coloane, numerotate de la **1** la **7**, având inițial toate elementele nule. Fără a utiliza alte variabile decât cele menționate, scrieți una sau mai multe instrucțiuni care pot înlocui puntele de suspensie, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat. (6p.)

```
for(i=1;i<=7;i++)
  for(j=1;j<=7;j++)
    .....
1 1 1 1 1 1 2
1 1 1 1 1 2 3
1 1 1 1 2 3 3
1 1 1 2 3 3 3
1 1 2 3 3 3 3
1 2 3 3 3 3 3
2 3 3 3 3 3 3
```

SUBIECTUL al III-lea**(30 de puncte)**

1. Subprogramul **conversie** are trei parametri:

- **n**, prin care primește un număr de cel mult 6 cifre scris în baza **b1**
- **b1**, prin care primește baza numărului original **n**
- **b2**, prin care primește baza în care va fi trecut numărul **n**

Subprogramul returnează numărul obținut în baza **b2**, acest număr va avea maxim 18 cifre.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=2020**, **b1=3** și **b2=4**, în baza 10, **n** va fi **60**, iar valoarea furnizată de subprogram este **330**.

(10 p.)

2. Un text are cel mult 200 de caractere, iar cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații. Scrieți un program Pascal/C/C++ care citește de la tastatură textul și care codifică acest text prin inversarea cuvintelor care au număr egal de vocale și consoane. Textul final va avea aceeași lungime cu textul inițial, spațiile libere dintre cuvinte se păstrează.

Exemplu: textul: **azi avem simulare la info**

Se va afișa textul: **azi meva eralumis al ofni**

(10 p.)

3. Fișierul **cifre.in** conține pe prima linie cel mult **10⁶** cifre. Să se citească cifrele din fișier și să se determine prima secvență cu valori identice de lungime maximă. Secvența va fi identificată prin poziția de început, poziția de final și valoarea cifrei din secvență. Afișarea se va face pe ecran. Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 2 0 0 1 2 2	2 0 1 9 2 0 2 0
Pe ecran	4 5 0	1 1 2

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

(2 p.)

b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat.

(8 p.)

REZOLVARE SUBIECTE
TEST 6
SUBIECTUL I

(20 de puncte)

1c	2d	3b	4c	5d	5x4p
----	----	----	----	----	------

SUBIECTUL al II – lea

(40 de puncte)

1. a. 237

 b. orice număr de 5 cifre care are **prima** cifră 1, **a treia** cifră 2 și **a cincea** cifră 3.
 (de exemplu **10203**)

```

c.
#include <iostream>
using namespace std;
int main()
{ int n,x;
  cin>>n;
  x=0;
  do
  { x=x*10+n%10;
    n=n/10;
  } while (n!=0);
  do
  { n=n*10+x%10;
    x=x/100;
  } while (x!=0);

  cout<<n;
  return 0;
}

```

```

d.
citește n (număr natural)
x←0
┌cât timp n≠0 execută
| x← x*10+n%10
| n←[n/10]
└─┘
┌cât timp x≠0 execută
| n← n*10+n%10
| x←[x/100]
└─┘
scrie n

```

2. `cin>>x[0].nr_mat>>x[0].data.zi>>x[0].data.luna>>x[0].data.an;`
`if (x[0].data.an<2013)`
`cout<<x[0].nr_mat;`
`else cout<<"imposibil";`
3. `if (i+j==8) a[i][j]=2;`
`else if (i+j<8) a[i][j]=1;`
`else a[i][j]=3;`

SUBIECTUL al III - lea**(30 de puncte)****1.****long long conversie(int n, int b1, int b2)**

```

{
    //formarea numarului din baza b1 in baza 10
    long long x=0,p=1,m;
    while(n)
    {
        x=x+(n%10)*p; //fiecare cifra se inmulteste cu b1^0, b1^1...
        p=p*b1;
        n=n/10;
    }
    //numarul x, il impartim la b2
    //cu resturile sale formam numar cu fixare de cifre in fata lui
    m=0; p=1;
    while(x)
    {
        m=m+(x%b2)*p;
        p=p*10;
        x=x/b2;
    }
    return m;
}

```

2.

```

#include<iostream>
#include<cstring>
using namespace std;
char s[202];
int i,p1,p2,l,j,ok;
int main()
{
    cin.get(s,201,'\n');
    l=strlen(s);
    for(i=0;i<l;i++)
    {
        //delimitam cuvintele-inceput
        if ((i==0 or s[i-1]==' ') and s[i]!=' ') p1=i;
        //finalul de cuvant
        if ((i==l-1 or s[i+1]==' ') and s[i]!=' ')
        {
            p2=i;
            //parcurgem intervalul [p1,p2] si calculam vocalele/consoanele
            ok=0;

```



```

for(j=p1;j<=p2;j++)
    if (strchr("aeiou",s[j])!=NULL) ok++;
if (ok*2==p2-p1+1)
    {
        ///dublul numarului de vocale este lungimea cuvântului, facem inversarea
        while(p1<p2)
        {
            ///interchimbare literele de la capete, spre mijlocul cuvântului
            swap(s[p1],s[p2]);
            p1++;
            p2--;
        }
    }
}
cout<<s;
}

```

3.

a. Se citesc numerele din fișier și în timpul citirii se compară numărul curent y cu cel anterior x. În caz de egalitate se va crește lungimea, iar în caz negativ înseamnă că am terminat o secvență. Fiecare secvență terminată în timpul citirii dar și la final de citire va fi testată pe baza lungimii ei, comparând-o cu variabila lmax, actualizând pe parcurs, poziția de început și final alături de valoarea din secvența maximă. Algoritmul este eficient ca timp de executare întrucât este liniar și este eficient ca spațiu de memorare întrucât nu folosește tablouri.

b.

```

#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("cifre.in");
int x,l,poz,xmax,lmax,st,dr,y;
int main( )
{
    fin>>x;
    ///prima secventa cu 1 element
    l=1;
    poz=1;
    lmax=0;
    while(fin>>y)
    {
        poz++;
        if (x==y) l++;
        else
        {
            ///s-a incheiat o secventa

```

```
    if (l>lmax)
    {
        lmax=l;
        xmax=x;
        st=poz-l;//poz minus l(variabila)
        dr=poz-1;//poz minus 1(unu, constanta)
    }
    //pregatim noua secv
    l=1;
    x=y;
}
}
//ultima secv se testeaza
if (l>lmax)
{
    lmax=l;
    xmax=x;
    st=poz-l+1;
    dr=poz;
}
cout<<st<<' '<<dr<<' '<<xmax;
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 6**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

Răspuns	Punctaj
1c 2d 3b 4c 5d	5x4p.

SUBIECTUL al II – lea**(40 de puncte)**

1.	a) Răspuns corect: 237	6p.	
	b) Răspuns corect: orice număr de 5 cifre care are prima cifră 1, a treia cifră 2 și a cincea cifră 3. (de exemplu 10203)	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se acordă numai 4p. dacă se folosesc tot structuri repetitive cu test final.
2.	Pentru rezolvare corectă - acces la câmpurile de pe primul nivel al înregistrării - acces la câmpurile de pe al doilea nivel al înregistrării - afișare conform condiției impuse(*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă - acces la un element al tabloului - atribuire a valorilor indicate elementelor tabloului (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*)Se acordă câte 1p. pentru atribuirea valorilor conform cerinței elementelor situate sub diagonala secundară, respectiv celor situate deasupra diagonalei secundare și 2p. pentru atribuirea valorilor conform cerinței elementelor situate pe diagonala secundară.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL al III - lea**(30 de puncte)**

1	Pentru subprogram corect - antet subprogram (*) - determinarea numerelor cerute (**) - declarare a tuturor variabilelor locale, corectitudine globală a subprogramului1)	10p. 3p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (tip, nume, parametri de intrare) conform cerinței. (**) Se acordă câte 3p. pentru: - Transformarea din baza b1 în baza 10 (1pct -obținerea cifrelor, 2 pct-formarea nr) - Transformarea din baza 10 în baza b2 (1pct -obținerea cifrelor, 2 pct-formarea nr)
2	Pentru program corect - declarare a unei variabile care să memoreze un șir de caractere - citirea textului - modificarea textului conform cerinței (*) - afișarea textului - declarare a variabilelor simple, - corectitudine globală a programului1)	10p. 1p. 1p. 5p. 1p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect - obținerea unui cuvânt - determinarea lungimii unui cuvânt, - determinarea nr de vocale - determinarea nr de consoane - inversarea unui cuvânt care respecta cerința
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este
	b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului1)	8p. 1p. 5p. 1p. 1p.	principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. Contorizam lungimea secvenței dacă numărul citit coincide cu anteriorul, la terminarea unei secvențe sau a întregii citiri, comparăm lungimea secvenței cu lungimea maximă. Nu se vor folosi vectori, citire și prelucrare în același timp. Complexitate liniară.

MODEL TEST 7

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **x** memorează un număr natural de exact 6 cifre. Care dintre următoarele instrucțiuni C/C++ elimină din **x** cele două cifre situate la mijlocul numărului?

- a. **`x/10000*100+x%100;`** b. **`x=x/100-x/100%100+x%100;`**
 c. **`x=x/10000+x%100;`** d. **`x%10000+x/100;`**

2. Se consideră **z** o variabilă globală care are valoarea **5** și următorul subprogram **f**:

```
int z=5;
void f(int x, int &y)
{ x++; y+=x; z+=y; }
```

Care va fi valoarea variabilei **z** după apelul **f(z,z)**?

- a. **16** b. **22** c. **5** d. **24**

3. Se consideră următoarele declarații:

```
struct Elev {char nume[30]; float mg;};
Elev E[100];
```

Care dintre următoarele variante reprezintă numele primului elev din vectorul **E**?

- a. **E[0].nume** b. **Elev[0].nume** c. **E.nume[0]** d. **Elev.E[0].nume**

4. Utilizând metoda backtracking, se generează toate secvențele formate din **n** caractere din mulțimea {'+', '-', 'o'}, astfel încât în orice prefix al unei secvențe astfel generate numărul de caractere '-' nu depășește numărul de caractere '+'. De exemplu, pentru **n=3**, s-au generat (în această ordine) următoarele secvențe:

+++ , ++o , ++- , +o+ , +oo , +o- , +-+ , +-o , o++ , o+o , o+- , oo+ , ooo.

Dacă utilizăm același algoritm pentru a genera secvențele de lungime **n=4**, care va fi cea de a opta secvență generată?

- a. **+++o** b. **+++-** c. **++-o** d. **++++**

5. Un graf neorientat are **50** de vârfuri și **5** componente conexe. Numărul maxim de muchii pe care le poate avea acest graf este:

- a. **225** b. **575** c. **2070** d. **1035**

SUBIECTUL II

(40 puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întreagă a numărului natural x .

- Scrieți valoarea afișată în urma executării algoritmului dacă se citesc numerele 1983 și 237. (6p.)
- Scrieți o pereche de numere care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze valoarea 11. (6p.)
- Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască a doua structură repetitivă cu o structură repetitivă alt de alt tip. (6p.)

citește a, b (numere naturale) $x \leftarrow 0$

```

execută
  x ← x*10+a%10+b%10
  a ← [a/10]
  b ← [b/10]
cât timp a=0 sau b=0

```

 $n \leftarrow 0$

```

repetă
  n ← n*10+x/10%10
  x ← [x/100]
până când x=0
scrie n

```

2. În declararea alăturată, variabila x memorează, pentru fiecare dintre cei 30 de elevi dintr-o clasă, numărul matricol, precum și data nașterii elevului. Scrieți o secvență de instrucțiuni C/C++ care citește de la tastatură informațiile despre elevii din clasă și afișează pe ecran numărul matricol al ultimului elev care are anul nașterii număr par, sau mesajul **nu există** în caz contrar. (6p.)

```

struct elev
{ int nr_mat;
  struct
  { int zi,luna,an;
  }data;
}x[30];

```

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 7 linii și 7 coloane, numerotate de la 1 la 7, având elemente de tip **char**. Fără a utiliza alte variabile decât cele menționate, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie, astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul alăturat. (6p.)

```

for(i=1;i<=7;i++)
  for(j=1;j<=7;j++)
    .....

    !####!
    ?!###!?
    ??#!??
    ???!???
    ??#!??
    ?!###!?
    !#####

```

SUBIECTUL III**(30 puncte)**

1. Subprogramul **radical** are patru parametri:
- **n**, prin care primește un număr natural nenul ($n \leq 10^9$)
 - **k**, prin care primește un număr natural mai mare sau egal cu 2
 - **a**, prin care furnizează numărul maxim extras din radicalul de ordin **k**, al numărului **n**
 - **b**, prin care furnizează numărul minim ce rămâne sub radicalul de ordinal **k**.

Scrieți definiția completă a subprogramului.

Exemplu: dacă $n=2020$ și $k=2$, atunci $\sqrt[2]{2020} = 2\sqrt{505}$, adică **a** va fi **2**, iar **b** va fi **505**.

(10 p.)

2. Un text are cel mult 200 de caractere, iar cuvintele sunt formate numai din litere mici ale alfabetului englez și sunt separate prin unul sau mai multe spații. Scrieți un program Pascal/C/C++ care citește de la tastatură textul și care codifică acest text prin transformarea literelor din cuvintele care au număr egal de vocale și consoane astfel: litera a va deveni litera z, litera b va deveni litera y, ... , litera z va deveni litera a. Textul final va avea aceeași lungime cu textul inițial, spațiile libere dintre cuvinte se păstrează.

Exemplu textul: **azi avem simulare la info**

Se va afișa textul: **azi zevn hrnfoziv oz rmul**

(10 p.)

3. Fișierul **cifre.in** conține pe prima linie cel mult 10^6 cifre. Să se citească cifrele din fișier și să se rearanjeze dacă este posibil astfel încât să formeze cel mai mare număr palindrom, sau mesajul **IMPOSIBIL**. Afișarea se va face pe ecran. Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 1 9 2 0 2 0	2 2 1 2 0 2 0
Pe ecran	IMPOSIBIL	2201022

- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2 p.)**
- b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat. **(8 p.)**

REZOLVARE SUBIECTE
TEST 7
SUBIECTUL I
(20 de puncte)

1b	2b	3a	4c	5d	5x4p
----	----	----	----	----	------

SUBIECTUL al II – lea
(40 de puncte)
1.

- a) După execuția primei structuri repetitive vom obține $x=1121$
 După execuția celei de-a doua structuri repetitive vom obține $n=21$
 Răspuns **21**
- b) Orice pereche de numere care să furnizeze după prima structură repetitivă valoarea x de forma **a1b1** va da răspunsul corect. O astfel de pereche este: $a=2131$ și $b=5050$

c) Programul C/C++:

```
#include <iostream>
using namespace std;
int main()
{
    int a,b, x, n=0;
    cin>>a>>b;
    x=0;
    do{
        x=x*10+a%10+b%10;
        a=a/10;
        b=b/10;
    }while(a!=0&& b!=0);
    if(x!=0)
    do{
        n=n*10+x/10%10;
        x=x/100;
    }while( x!=0);
    cout<<n;
    return 0;}

```

d) Algoritmul în pseudocod echivalent care înlocuiește a doua structură repetitivă este:

```
citeste a,b (numere naturale)
x←0
execută
    x← x*10 + a%10+b%10
    a← [a/10]
    b← [b/10]
■cat timp a≠0 și b≠0
n←0
    cat timp x≠0 executa
        n←n*10+x/10%10
        x← [x/100]
    ■
scrie n

```

2. Secvența de program care rezolvă cerința:

```
int poz = -1; // poz = poziția ultimului copil al cărui an de naștere este număr par
for( int i=0 ; i<30 ; i++)
{
    cin>>x[i].nr_mat>>x[i].data.zi>>x[i].data.luna>>x[i].data.an;
    if( x[i].data.an %2 ==0) poz = i;
}
if(poz == -1) cout<<"nu exista";
else cout<<x[poz].data.nr_mat;

```


3. Instrucțiunile care ar putea fi scrise în locul punctelor de suspensie pentru a construi matricea din exemplu, sunt:

```
if( i==j || i+j==8) a[i][j]='!';
else if( i<j && i+j<8 && i<4 || i>j && i+j>8 && i>4) a[i][j]='#';
    else a[i][j]='?';
```

SUBIECTUL al III – lea
(40 de puncte)

1. **void radical(int n,int k, int &a, int &b)**

```
{
    ///descompunem in factori primi
    a=1; b=1;
    int e,i,d=2;
    while(n>1)
    {
        e=0;
        while(n%d==0)
        {
            e++;
            n=n/d;
        }
        /// modificarea lui a
        for(i=1;i<=e/k;i++)
            a=a*d;
        /// modificarea lui b
        for(i=1;i<=e%k;i++)
            b=b*d;
            d++;
    }
}
```

2. **#include<iostream>**

```
#include<cstring>
```

```
using namespace std;
```

```
char s[202];
```

```
int i,p1,p2,l,j,ok;
```

```
int main()
```

```
{ cin.get(s,201,'\n');
```

```
l=strlen(s);
```

```
for(i=0;i<l;i++)
```

```
{
```

```
///delimitam cuvintele- p1 este indicele de inceput al cuvântului utilizat din text
```

```
if ((i==0 or s[i-1]==' ') and s[i]!=' ') p1=i;
```

```
///p2- este indicele de finalul al cuvântului
```

```
if ((i==l-1 or s[i+1]==' ') and s[i]!=' ')
```

```
{
```

```
p2=i;
```

```
///parcurem caracterele din cuvânt de la indicii situați în intervalul [p1,p2] și
```

```
///calculăm numărul de vocale
```

```
ok=0;
```

```
for(j=p1;j<=p2;j++)
```

```
if (strchr("aeiou",s[j])!=NULL) ok++;
```

```
if (ok*2==p2-p1+1)
```

```

    {
        //dacă dublul numărului de vocale este egal cu lungimea cuvântului, facem
        // modificarea literelor
        for(j=p1;j<=p2;j++)
            s[j]='a'+(25- (s[j]-'a'));
    }
}
}
cout<<s; }

```

3. a. Fiind doar cifre în fișierul de intrare, vom defini un vector de frecvență de dimensiune 10. Pentru fiecare cifră vom stabili numărul ei de apariții. Problema nu va avea soluție dacă există 2 sau mai multe cifre cu frecvențe impare. Dacă există soluție, cifrele se vor afișa în două seturi, folosind în fiecare set un număr de cifre egal cu jumătate din frecvența fiecărei cifre. Palindromul de valoare maximă va începe cu un set de cifre alese în sens descrescător, va continua cu cifra cu frecvență impară (dacă există) și se va termina cu al doilea set de cifre alese în sens crescător.

b. **#include<fstream>**

#include<iostream>

using namespace std;

ifstream fin("cifre.in");

int x,f[10],c,imp,i;

int main()

{

while(fin>>x)

f[x]++;

imp=0;

for(c=0;c<=9;c++)

if (f[c]%2==1) imp++;

if (imp>1) cout<<"IMPOSIBIL";

else

{

///af în ordine descrescătoare cifrele, fiecare fiind scrisă de jumătate din frecvența sa

for(c=9;c>=0;c--)

for(i=1;i<=f[c]/2;i++)

cout<<c;

///verificam daca avem o frecvență impară

for(c=0;c<=9;c++)

if (f[c]%2==1) cout<<c;

///af în ordine crescătoare cifrele, fiecare fiind scrisă de jumătate din frecvența sa

for(c=0;c<=9;c++)

for(i=1;i<=f[c]/2;i++)

cout<<c;

}

}

BAREM DE EVALUARE ȘI DE NOTARE**TEST 7**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 puncte)**

Răspuns	Punctaj
1b 2b 3a 4c 5d	5x4p.

SUBIECTUL II**(40 puncte)**

1.	a) Răspuns corect: 21	6p.	
	b) Răspuns corect: orice pereche de numere care furnizează rezultatul final 11	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are a doua structură repetitivă conform cerinței, principal corectă, dar nu este echivalent cu cel dat.
2.	Pentru rezolvare corectă - acces la câmpurile de pe primul nivel al înregistrării - acces la câmpurile de pe al doilea nivel al înregistrării - afișare conform condiției impuse(*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă - acces la un element al tabloului - atribuire a valorilor indicate elementelor tabloului (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru atribuirea valorilor conform cerinței, pentru elementele situate între cele două diagonale și 2p. pentru atribuirea valorilor elementelor situate pe diagonala secundară și pe diagonala principală.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL III**(30 puncte)**

1	Pentru subprogram corect - antet subprogram (*) - determinarea numerelor cerute (**) - declarare a tuturor variabilelor locale, corectitudine globală a subprogramului)	10p. 3p. 6p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametrii de intrare, parametrii de ieșire) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (factori primi și exponenți + modificare variabila a + modificare variabila b)
2	Pentru program corect - declarare a unei variabile care să memoreze un șir de caractere - citirea textului - modificarea textului conform cerinței (*) - afișarea textului - declarare a variabilelor simple, - corectitudine globală a programului)	10p. 1p. 1p. 5p. 1p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect - obținerea unui cuvânt - determinarea lungimii unui cuvânt, - determinarea nr de vocale - determinarea nr de consoane - codificarea unui cuvânt care respecta cerința
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului)	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă este folosirea vectorilor de frecvență. Cazul IMPOSIBIL , e îndeplinit dacă avem 2 sau mai multe cifre cu frecvențe impare. Soluția va fi obținută prin afișarea descrescătoare a cifrelor și afișarea crescătoare a cifrelor, în fiecare parte punând jumătate din câte am citit. Caz special ce cifra va fi la mijloc, dacă avem 1 singură frecvență impară?

MODEL TEST 8

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

Subiectul I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **n** este de tip întreg. Care dintre următoarele expresii C/C++ are valoarea 1 dacă și numai dacă valoarea memorată de **n** este divizibilă cu 3 și nenulă?

a) **`n%10==3`** b) **`n/3==0`** c) **`(n-3)%3==0`** d) **`n%2==1`**

2. Funcția **f** are definiția alăturată. Ce se va afișa la apelul **f(4)**?

```
void f(int n)
{ int i;
  for(i=1;i<=n;i++)
  { if(i%2==1)f(i-1);
    cout<<i;
  }
}
```

a) 1214 b) 121234 c) 112234 d) 123412

3. Utilizând metoda backtracking se generează toate șirurile de 4 valori din mulțimea {1,2,3,4,5} astfel încât pe oricare două poziții alăturate să nu se afle două valori de aceeași paritate. Primele 7 soluții generate sunt: {1,2,1,2}, {1,2,1,4}, {1,2,3,2}, {1,2,3,4}, {1,2,5,2}, {1,2,5,4}, {1,4,1,2}. Care este a 8-a soluție?

a) {2,1,3,1} b) {1,4,3,2} c) {2,3,2,1} d) {1,4,1,4}

4. Se consideră un graf cu 11 vârfuri și 55 de muchii. Numărul minim de muchii care trebuie eliminate din graf astfel încât să se obțină un graf eulerian este:

a) 10 b) 11 c) 0 d) 1

5. Un graf orientat are 7 noduri numerotate de la 1 la 7 și arcele: (1,2), (2,3), (4,5), (5,6), (6,4), (7,5). Care este numărul minim de arce care trebuie adăugate astfel încât graful să devină tare conex?

a) 1 b) 2 c) 3 d) 0

Subiectul II**(40 puncte)**

Scrieți pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

1. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .

- Scrieți ce valoare se va afișa, dacă se citesc valorile **895** și **124**. (6p.)
- Scrieți toate perechile de numere ce pot fi citite pentru **a** și **b**, cu $a < b$, astfel încât să se afișeze valoarea **2020**. (6p.)
- Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)
- Scrieți în pseudocod un algoritm echivalent cu cel dat, care să nu cuprindă nicio structură repetitivă. (6p.)

citește **a** și **b** (numere naturale nenule, a și $b < 10^5$)

p=1;e=0;

cat timp b≠0 executa

p1=1;x=b%10;t=0;c=0;aux=a;

cat timp aux≠0 executa

y=aux%10

d=y*x+t

t=[d/10]

d=d%10

c=c+d*p1;

p1=p1*10;

aux=[aux/10];

daca t≠0 atunci

c=c+t*p1;

e=e+c*p; p=p*10; b=[b/10]
scrie e;

2. Variabilele **i** și **j** sunt de tip întreg, iar variabila **a** memorează un tablou bidimensional cu **6** linii și **6** coloane, numerotate de la **0** la **5**, având inițial toate elementele egale cu valoarea **-1**. Fără a utiliza alte variabile, completați secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila **a** să memoreze tabloul alăturat. (6p)

```
for(i=0;i<6;i++)
```

```
for(j=0;j<6;j++)
```

```
.....
5   1  2  3  4  0
1   4  1  2  0  4
2   1  3  0  2  3
3   2  0  2  1  2
4   0  2  1  1  1
0   4  3  2  1  0
```

3. Variabila **s** memorează simultan următoarele date despre fiecare dintre cele **40** de clase din cadrul unei școli: un cod de maximum **5** caractere, reprezentând numele clasei, numărul de elevi din clasă și mediile acestora. În fiecare clasă sunt maximum **30** de elevi. Știind că expresiile **C/C++** de mai jos au ca valori un șir de caractere ce reprezintă codul celei de a **2**-a clase, o valoare naturală ce reprezintă numărul de elevi din cea de a **2**-a clasă, respectiv o valoare reală ce reprezintă media celui de al **3**-lea elev din a **2**-a clasă, scrieți definiția unei structuri cu eticheta **clasa**, care permite memorarea datelor despre o clasă, și declarați corespunzător variabila **s**. (6p.)

s[1].cod **s[1].NrElevi** **s[1].Medie[2]**

SUBIECTUL III**(30 puncte)**

1. Subprogramul **inserare** are doi parametri:

- **n**, prin care primește un număr natural ($n \in [10, 10^5]$);
- **d**, prin care furnizează numărul obținut prin inserarea între două cifre de aceeași paritate lui **n**, media lor aritmetică sau **-1** dacă acesta nu conține două cifre de aceeași paritate alăturate.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=1976**, după apel **d=159876**, iar pentru **n=1234**, după apel **d=-1**. (10p.)

2. Un text are cel mult **255** de caractere, iar cuvintele sale sunt formate numai din litere mici ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program **C/C++** care citește de la tastatură un text de tipul precizat mai sus, și afișează pe ecran șirul modificat prin eliminarea din textul inițial a acelor cuvinte în care vocalele sunt ordonate strict lexicografic. Un cuvânt format doar din consoane va fi eliminat din text. Dacă textul inițial nu conține cuvinte care îndeplinesc condiția cerută se va afișa pe ecran doar mesajul **nu exista**. Se consideră vocale literele **a, e, i, o, u**.

Exemplu: pentru textul **ei aduc multe carti**, se va afișa textul **multe**, iar pentru textul **ea aduce multa bucurie** se va fișa mesajul **nu exista**. (10p.)

3. Șirul **f** este definit astfel: $f_1=1$, $f_2=4$, $f_i=2*(f_{i-1}+1)-f_{i-2}$, $i \geq 3$. Fișierul **bac.txt** conține pe prima linie un număr natural **n** ($1 \leq n \leq 10^9$), iar pe a doua linie cel mult 10^9 numere naturale din intervalul $[1, 100^2]$. Se cere să se afișeze pe ecran numărul care ar apărea pe poziția **n** în șirul ordonat crescător obținut din toate numerele aflate pe a doua linie a fișierului care fac parte din șirul **f**. Dacă pe ultima linie a fișierului sunt mai puțin de **n** termeni ai șirului menționat anterior se afișează pe ecran mesajul **Nu exista**. Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul **bac.txt** conține numerele:

7

4 1 34 4 2 3 81 121 5 7 8 1 1 12 169 15 9 1 24

atunci se va afișa pe ecran valoarea **9**.

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. (2p.)

b) Scrieți programul **C/C++** corespunzător algoritmului descris. (8p.)

REZOLVARE SUBIECTE

TEST 8

Subiectul I

1. c
2. b
3. d
4. c
5. b

Subiectul II

1.

a) 110980

b) Perechile cerute sunt: 1 și 2020, 2 și 1010, 4 și 505, 5 și 404, 10 și 202, 20 și 101

c) Programul C/C++

```
#include <iostream>
using namespace std;
int main()
{
    int a,b,c,d,e,p,p1,aux,t,x,y;
    cin>>a>>b;
    p=1;e=0;
    while(b!=0)
    {
        p1=1;x=b%10;aux=a;t=0;c=0;
        while(aux!=0)
        {
            y=aux%10;
            d=y*x+t;
            t=d/10;
            d=d%10;
            c=c+d*p1;
            p1=p1*10;
            aux=aux/10;
        }
        if(t!=0)c=c+t*p1;
        e=e+c*p;
        p=p*10;
        b=b/10;
    }
    cout<<e;
    return 0;
}
```

d) Algoritmul calculează produsul celor două numere citite.
citește a și b (numere naturale nenule, a și b <10⁵)
scrie a*b.

2.

```

for(i=0;i<6;i++)
  for(j=0;j<6;j++)
    if(i==j)a[i][j]=5-i;
    else if(i+j==5)a[i][j]=0;
    else if(j>i)a[i][j]=j-i;
    else a[i][j]=i-j;

```

3.

```

struct clasa
{
  char cod[5];
  int NrElevi;
  float Medie[30];
}s[40];

```

Subiectul III

1. Comparăm câte două cifre alăturate din număr.

```

void inserare(int n, int &d)
{
  int c1,c2,p,n1;
  n1=n;p=10;
  c1=n%10;d=c1;
  n=n/10;
  while(n!=0)
  {
    c2=n%10;
    n=n/10;
    if(c1%2==c2%2)
    {
      d=d+(c1+c2)/2*p;
      p=p*10;
    }
    d=d+c2*p;
    p=p*10;
    c1=c2;
  }
  if(d==n1)d=-1;
}

```

2. Extragem pe rând câte un cuvânt din text și verificăm dacă vocalele sunt ordonate strict crescător. În caz afirmativ îl eliminăm din șir. O posibilă soluție este următoarea.

```

#include <iostream>
#include <string.h>
using namespace std;

int main()
{
  char s[255],voc[]="aeiou",*p,*q,cuv[100];
  int i,poz,ok,gasit=0;
  cin.get(s,255);
  p=s;q=strchr(s, ' ');
  while(q!=NULL)

```

```

{
    cuv[0]=0;
    strncat(cuv,p,q-p);
    poz=-1;ok=1;
    for(i=0;i<strlen(cuv);i++)
        if(strchr(voc,cuv[i])!=NULL)
            if(poz==-1)poz=i;
            else if(cuv[i]<=cuv[poz])ok=0;
            else poz=i;
    if(ok==1)
    {
        strcpy(p,q);gasit=1;p=p+1;
    }
    else p=q+1;
    q=strchr(p, ' ');
}
poz=-1;ok=1;
for(i=0;i<strlen(p);i++)
    if(strchr(voc,p[i])!=NULL)
        if(poz==-1)poz=i;
        else if(p[i]<=p[poz])ok=0;
        else poz=i;
if(ok==1){*p=NULL;gasit=1;}
if(gasit==1)cout<<s;
else cout<<"nu exista";
return 0;
}

```

3.

- a) Se observă faptul că valorile din șir sunt pătrate perfecte. O soluție eficientă utilizează un vector de frecvență cu 100 elemente cu ajutorul căruia vom memora de câte ori a apărut valoarea k . Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determinăm fiecare valoare din șir de câte ori apare; în funcție de numărul de apariții a fiecărei valori vom identifica valoarea cerută. Complexitatea algoritmului depinde doar de numărul de valori din fișier.

- b) O posibilă soluție este următoarea:

```

#include <iostream>
#include <fstream>
#include <cmath>
using namespace std;
ifstream fin("bac.txt");
int ap[101];
int main()
{
    int k,n,x;
    fin>>n;
    while(fin>>k)
    {
        x=floor(sqrt(k));
        if(x*x==k)ap[x]++;
    }
    x=1;
    while(x<=100&&ap[x]>0)
    {
        n=n-ap[x];
        x++;
    }
    if(n<=0)cout<<(x-1)*(x-1);
    else cout<<"nu exista";
    return 0;
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 8**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

1. c	2. b	3. d	4. c	5. b	5x4p.
-------------	-------------	-------------	-------------	-------------	--------------

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 110980	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte un punct pentru fiecare dintre perechile: 1 și 2020, 2 și 1010, 4 și 505, 5 și 404, 10 și 202, 20 și 101
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiune de decizie - instrucțiuni repetitive (*) - atribuirii - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 1p. 4p. 1p. 1p.	(*) Se acordă numai 2p dacă doar una dintre instrucțiunile repetitive este conform cerinței
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	citește a,b scrie a*b
2.	Pentru rezolvare corectă*	6p.	(*) Se acordă câte 1p. pentru fiecare aspect: (determinare valori de pe diagonala principală, diagonala secundară, zona de nord a matricei, zona est, zona sud, zona vest)
3.	Pentru rezolvare corectă	6p.	(*) Se acordă câte 2 puncte pentru fiecare aspect al cerinței (declare corectă pentru fiecare câmp al structurii)

SUBIECTUL al III - lea**(30 de puncte)**

1.	Pentru subprogram corect - antet subprogram (*) - determinare a numărului cerut (**) - declarare a variabilelor locale - corectitudine globală a subprogramului ¹⁾	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (testarea parității a două cifre alăturate, calcularea mediei aritmetice, inserarea unei cifre în număr).
2.	Pentru program corect	10p	(*) Se acordă câte 2p pentru fiecare aspect al

	<ul style="list-style-type: none"> - declarare a unei variabile care să memoreze un șir de caractere - citire a datelor - determinare șirului cerut (*) - afișare a datelor - declarare a variabilelor simple, corectitudine globală a programului¹⁾ 	<p>1p</p> <p>1p</p> <p>6p</p> <p>1p</p> <p>1p</p>	<p>cerinței (identificare cuvânt, verificare poziționare vocale, eliminare cuvânt din text)</p>
3.	<p>a) Pentru răspuns corect</p> <ul style="list-style-type: none"> - coerența descrierii algoritmului (*) - justificare a unor elemente de eficiență 	<p>2p.</p> <p>1p.</p> <p>1p.</p>	<p>(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.</p>
	<p>Pentru program corect</p> <ul style="list-style-type: none"> - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*), (**) - utilizarea unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului¹⁾ 	<p>8p.</p> <p>1p.</p> <p>5p.</p> <p>1p.</p> <p>1p.</p>	<p>(**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare.</p> <p>(***) Se acordă punctajul numai pentru un algoritm liniar</p> <p>Se observă faptul că valorile din șir sunt pătrate perfecte. O soluție eficientă utilizează un vector de frecvență cu 100 elemente cu ajutorul căruia vom memora de câte ori a apărut valoarea k. Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determinăm numărul de apariții a fiecăre valori din șir, complexitatea algoritmului depinde doar de numărul de valori din fișier.</p>

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

MODEL TEST 9

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I
(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Dacă variabilele a și b sunt de tip `int`, ce valori vor avea variabilele a și b la finalul executării secvenței de instrucțiuni alăturate:

- a. $a=0$ și $b=50$
 b. $a= -1$ și $b=30$

```
a=5; b=0;
do{
a--; b+=a*a;
} while(a>0);
c.  $a=-1$  și  $b=31$ 
d.  $a=0$  și  $b=30$ 
```

2. Se consideră subprogramul cu definiția alăturată. Ce valoare se va afișa în urma executării instrucțiunii de mai jos?
`cout<<f(10); | printf("%d",f(10));`

a. 01010

b. 1010

```
int f (int n)
{ int c;
if (n!=0)
{if (n%2==1)
    c=1+f(n/2);
    else c=f(n/2);
    cout<<n%2; | printf("%d",n%2);
    return c;
}
else return 0; }
```

c. 10102

d. 21010

3. Un număr este palindrom dacă citit de la stânga la dreapta se obține același număr. Generând palindroamele de lungime 3, folosind cifrele 0, 1, 2, 3, 4 se obțin în ordine numerele: 101, 111, 121, 131, 141, 202, 212, Folosind același algoritm pentru a genera toate palindroamele pare de lungime 4 folosind cifrele 0, 1, 2, 3, 4, 5 care este al șaptelea număr generat?

a. 2002

b. 2442

c. 4004

d. 4224

4. Care este numărul grafurilor orientate cu n noduri cu proprietatea că pentru orice pereche de noduri distincte i și j există cel puțin un arc între i și j .

a. 3^n

b. n^3

c. $3^{n*(n-1)}$

d. $3^{n*(n-1)/2}$

5. Se consideră un arbore cu 8 noduri. Lista alăturată reține pentru fiecare nod al arborelui fiii lui (descendenții direcți). Câte dintre nodurile lui ar putea fi alese ca rădăcină astfel încât arborele să aibă număr maxim de niveluri?

1: -
2: -
3: 2, 6, 8
4: -
5: 1
6: 4, 7
7: 5
8: -

a. 4

b. 3

c.2

d.1

SUBIECTUL al II-lea**(40 puncte)**

1. Algoritm alăturat este reprezentat în pseudocod. S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b .

a. Ce se va afișa dacă se citește pentru n valoarea 5 și pentru x valorile: 16, 80, 48, 20, 240 **(6p)**

b. Dacă $n=4$, dați exemplu de patru valori pentru x pentru care algoritmul să afișeze 2020. **(6p)**

c. Scrieți programul C/C++ corespunzător algoritmului. **(10p)**

d. Scrieți un algoritm în pseudocod echivalent cu algoritmul dat în care să se utilizeze doar structuri repetitive condiționate posterior (cu test final) **(6p)**

```

citește n (număr natural)
d ← 0
pentru i ← 1, n execută
    citește x (număr natural nenul)
    dacă d = 0 atunci
        d ← x
    altfel
        repetă
            r ← x % d
            x ← d
            d ← r
        până când r = 0
        d ← x
Scrie d

```

2. Variabila p memorează simultan informații referitoare la cei 100 angajați ai unei companii: numărul de identificare (un număr natural), numele (un șir cu maxim 50 caractere), salariul (un număr real), data nasterii și data angajării (ziua, luna și anul numere naturale). Știind că expresiile C/C++ de mai jos au ca valori id-ul, prima literă a numelui primului angajat, luna nașterii și anul angajării acestuia, scrieți definiția unei structuri cu eticheta **angajat**, care permite memorarea datelor despre angajații companiei, și declarați corespunzător variabila p .

$p[0].Id$ $p[0].Nume[0]$ $p[0].Data_N.luna$ $p[0].Data_A.an$ **(6p)**

3. Variabilele i și j sunt de tip întreg, iar variabila a memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 0 la 5, având inițial toate elementele egale cu valoarea 0. Fără a utiliza alte variabile, completați secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila a să memoreze tabloul de mai jos.

```

1  1  1  2  2  2
3  1  1  2  2  3
3  3  1  2  3  3
3  3  1  2  3  3
3  1  1  2  2  3
1  1  1  2  2  2

```

```

for(i=0;i<6;i++)
    for(j=0;j<6;j++)

```

.....

SUBIECTUL al III-lea

(30 puncte)

1. Scrieți definiția completă a unui subprogram **aranjare** care are doi parametri: **a** prin care primește un tablou unidimensional cu maximum 100 de numere întregi de maxim 4 cifre și **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile de 2 cifre să fie ordonate descrescător, celelalte elemente din vector nefiind afectate de modificări. Tabloul modificat va fi furnizat tot prin intermediul parametrului **a**. Scrieți definiția completă a subprogramului **aranjare**. (10p.)

Exemplu: dacă tabloul are 6 elemente și este de forma (12, -7, 61, -32, 800, 7), după apel, acesta va fi: (61, -7, 12, -32, 800, 7).

2. Spunem că un cuvânt **t** este derivat din cuvântul **s** dacă **s** apare o singură dată în cuvântul **t**, **s** este prefix al lui **t** și **t** are cel puțin un caracter în plus față de **s**. De exemplu, cuvântul **carte** este derivat din cuvântul **car** dar cuvântul **caricatura** nu este derivat din cuvântul **ca**. Un text are cel mult 100 de caractere și este format din cuvinte separate prin unul sau mai multe spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program C/C++ care citește de la tastatură textul și afișează pe ecran, separate între ele printr-un singur spațiu, cu majuscule cuvintele din text care derivă din primul cuvânt. Dacă textul nu conține nici un astfel de cuvânt se va afișa mesajul **NU EXISTA**.

Exemplu: pentru textul **el este acel elev care a fost eliminat iar ele sunt colegele lui** se va afișa pe ecran: **ELEV ELIMINAT ELE**

(10p.)

3. Se consideră șirul 1, 1, 2, 1, 2, 3, 1,2, 3, 4... .. construit astfel: prima grupă este formată din numărul 1, a doua grupă este formată din numerele 1 și 2, iar grupa a **k**-a, este formată din numerele 1, 2,**k**-1, **k**. Se cere să se citească din fișierul **bac.in** un număr natural **n** ($n \leq 10000$) și să se afișeze în fișierul **bac.out** cel de al **n**-lea termen al șirului dat. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă valoarea lui **n** este 10 se va afișa 4; dacă valoarea lui **n** este 12 se va afișa 2

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b) Scrieți programul C/C++ corespunzător algoritmului descris. (8p.)

REZOLVARE SUBIECTE
TEST 9
SUBIECTUL I

(20 de puncte)

1d	2c	3c	4d	5b	5x4p
----	----	----	----	----	------

SUBIECTUL al II – lea

(40 de puncte)

1. a. 4

b. oricare 4 numere al căror cmmdc este 4. Un răspuns corect ar putea fi: 2020, 6060, 10100, 14140

c.

```
#include <iostream>
using namespace std;
int main()
{ int n,d,x,r;
  cin>>n;
  d=0;
  for(int i=1;i<=n;i++)
  { cin>>x;
    if(d==0)
      d=x;
    else
    {
      do{
        r=x%d;
        x=d;
        d=r;
      } while(r);
      d=x;
    }
  }
  cout<<d;
  return 0;
}
```

d.

```
citește n (număr natural)
d←0
i←1
dacă i<=n atunci
  repetă
    citește x (număr natural nenul)
    dacă d=0 atunci
      d←x
    altfel
      repetă
        r←x % d
        x←d
        d←r
      până când r=0
    d←x
  i←i+1
  până când i>n
scrie d
```



```

2. struct angajat{
    unsigned int Id;
    char Nume[51];
    double salariu;
    struct {
        unsigned int zi, luna, an;
    } Data_N, Data_A;
} p[100];

```

```

3. for(i=0;i<6;i++)
    for(j=0;j<6;j++)
        if((i<=j && i+j<=5) || (i>=j && i+j>=5))
            //zona N sau zona S, inclusiv diagonalele
                if (j<=2)
                    a[i][j]=1;
                else
                    a[i][j]=2;
                else
                    a[i][j]=3;//zonele E sau V

```

SUBIECTUL al III - lea

(30 de puncte)

```

1. void aranjare(int a[], int n)
{
    for(int i=1;i<n;i++)
        if(a[i]/100==0&& a[i]/10!=0)
            for(int j=i+1;j<=n;j++)
                if(a[j]/100==0&& a[j]/10!=0)
                    if(a[i]<a[j])
                        swap(a[i],a[j]);
}

```

```

2.
#include <iostream>
#include <cstring>
using namespace std;
int main()
{
    int i,n,gasit=0;
    char s[101],*p, cuv[101];
    cin.getline(s,101);
    p=strtok(s, " ");
    strcpy(cuv,p); n=strlen(cuv); //se reține în cuv primul cuvânt
    while(p)
    {
        if(strstr(p,cuv)==p && strstr(p+n,cuv)==0 && strlen(p)!=strlen(cuv)) //daca p derivă din cuv
        {
            for(int i=0;i<strlen(p);i++) //se transformă toate literele lui p în majuscule (se știe că toate
            sunt litere mici)
                p[i]=p[i]-32;
            cout<<p<<' ';
            gasit=1;//se afișează cuvântul și se marchează că s-a găsit un cuvânt derivat din cuv
        }
        p=strtok(NULL, " ");
    }
    if(!gasit) cout<<"NU EXISTA";
    return 0;
}

```

3. Se împarte șirul 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, în grupe:

grupa 1: 1;

grupa 2: 1, 2;

grupa 3: 1, 2, 3;

.....

grupa k: 1, 2, ..., k;

- Presupunem că termenul de rang n este ultimul din grupa completă k; în acest caz se obține relația $1+2+3+\dots+k=n$, adică $k*(k+1)/2=n$
- Numărul de grupe complete ale șirului până la termenul de rang n se obține ca soluție a ecuației $k^2+k-2*n=0$
- Se verifică dacă termenul de rang n este ultimul dintr-o grupă completă sau face parte dintr-o grupa incompletă
- Se determină poziția termenului de rang n în cadrul grupei lui și se afișează, aceasta fiind și valoarea termenului cerut

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main()
```

```
{ long long n,d,k,p;
```

```
cin>>n;
```

```
d=1+8*n;// se determină discriminantul ecuației de gradul II corespunzătoare șirului
```

```
k=(-1+sqrt(d))/2; // se determină numărul de grupe complete până la termenul de rang n
```

```
if (n==k*(k+1)/2) //dacă termenul de rang n este ultimul din grupa completă k se afișează
```

```
cout<<k;
```

```
else
```

```
{
```

```
p=n-k*(k+1)/2; //se stabilește poziția termenului de rang n în cadrul grupei lui
```

```
cout<<p; //valoarea termenului este egală cu poziția lui în cadrul grupei lui
```

```
}
```

```
return 0;
```

```
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 9**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 puncte)**

1d	2c	3c	4d	5b	5x4p
-----------	-----------	-----------	-----------	-----------	-------------

SUBIECTUL II**(40 puncte)**

1.	a) Răspuns corect: 4	6p	
	b) Răspuns corect: oricare 4 numere al căror cmmdc este 2020	6p	Un răspuns corect ar putea fi: 2020, 6060, 10100, 14140
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie -instrucțiuni repetitive -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 1p. 4p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este corectă.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -definire a structurii/înregistrării (*) -declarare a variabilei conform cerinței -corectitudine globală a secvenței ¹⁾	6p. 3p. 2p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (definire principial corectă a unei structuri)
3.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (atribuire valori 1, atribuire valori 2, atribuire valori 3, elemente suport) conform cerinței.

SUBIECTUL al III - lea**(30 puncte)**

1.	Pentru subprogram corect -antet subprogram (*) -aranjare a elementelor în ordinea cerută (**) -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului	10p. 3p. 5p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, cei 2 parametri) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect specific (identificare a elementelor cu exact 2 cifre, oronare descrescătoare, pastrarea celorlalte elemente nemodificate)
2.	Pentru program corect -declarare a unei variabile care să memoreze un șir de caractere, declarare a variabilelor simple -citire a datelor -determinare a valorilor cerute (*) -afișare a datelor (**) - corectitudine globală a programului	10p. 1p. 1p. 4p. 3p. 1p.	(*) Se acordă câte 2p. pentru identificare corectă a entităților/cuvintelor, identificare corectă a cuvintelor derivate. (**) Se acordă câte 1p. pentru afișare unui cuvântul derivat, 1p afișare cu majuscule, 1p pentru mesajul NU EXISTA
3.	a) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm care nu folosește nicio structură repetitivă și care folosește doar variabile simple. O soluție posibilă citește valoarea lui n din fișier, stabilește numărul de grupe complete ale șirului, grupa din care face parte termenul de rang n, poziția termenului de rang n în cadrul grupei lui și valoarea termenului cerut Un algoritm posibil de rezolvare este: citește n $d \leftarrow 1+8*n$ $k \leftarrow (-1+\text{sqrt}(d))/2$ dacă $n=k*(k+1)/2$ atunci scrie k altfel $p \leftarrow n- k*(k+1)/2$ scrie p ■

MODEL TEST 10

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. În secvența de program de mai jos, instrucțiunea de afișare se va executa de un număr de ori egal cu:

```
for (i=1;i<=4;i++);
  for (j=i+1;j<=10;j++)
    cout<<j; | printf("%d",j);
```

a. 24

b. 6

c. 5

d. 30

(4 p)

2. Subprogramul f este definit alăturat. Indicați ce se afișează în urma apelului **f(12345)**.

```
void f (int x)
{ cout<<" "; | printf("+");
  if(x>0) { f(x/100);
  cout<<x; | printf("%d",x); }
  cout<<" "; | printf("+");
}
```

a. +++++1+123+12345+

c. ++++112312345

b. +12345+123+1+++++

d. +1+123+12345+

(4 p)

3. Utilizând metoda backtracking, sunt generate în ordine lexicografică toate anagramele cuvântului **house** astfel încât două vocale să nu fie alăturate. Să se precizeze câte anagrama se generează între soluțiile **esuho** și **osehu**.

a) 3

b) 0

c) 2

d) 1

(4 p)

4. Precizați care dintre următoarele grafuri orientate cu patru vârfuri date prin vectorul de arce este tare conex.

a) $U = ((1, 2), (1, 4), (3, 2), (3, 1))$ b) $U = ((1, 2), (1, 4), (2, 3), (3, 1), (4, 3))$ c) $U = ((1, 2), (1, 4), (1, 3))$ d) $U = ((1, 2), (2, 4), (4, 1))$

(4p)

5. Se consideră graful neorientat cu 7 noduri, reprezentat prin următoarea listă de muchii: [1, 7], [1, 2], [2, 7], [7, 6], [2, 6], [2, 5], [3, 4], [2, 3], [5, 6].

Să se precizeze numărul minim de muchii ce trebuie adăugate astfel încât graful să devină eulerian.

a) 3

b) 2

c) 4

d) 0

(4 p)

SUBIECTUL al II-lea**(40 de puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[c]$ partea întreagă a numărului real c .

- a) Scrieți ce valoare va fi afișată în urma execuției algoritmului dacă se citesc, în această ordine, valorile 11, 57, 9. (6 p)
- b) Dacă pentru z se citește valoarea 17, dați un exemplu de valori pentru a și b astfel încât să se afișeze două valori nule 0 0. (6 p)
- c) Să se scrie programul C/C++ corespunzător algoritmului dat. (10 p)
- d) Să se scrie un algoritm echivalent cu cel dat în care să nu se folosească structuri repetitive. (6p)

Citește a, b, c (numere naturale nenule)┌dacă $a > b$ atunci| $a \leftarrow a + b$ | $b \leftarrow a - b$ | $a \leftarrow a - b$

└─

 $s \leftarrow 0$ $n \leftarrow 0$ ┌pentru $x \leftarrow b, a, -1$ execută| ┌dacă $x \% c = 0$ atunci| | $n \leftarrow n + 1$ | | $s \leftarrow s + x$

| └─

└─

Scrie $n, ', s$

2. Se consideră următoarea declarație:

```
struct cerc
{ float r;
  struct
  {int x,y;}centru;
  }c1,c2;
```

Scrieți o expresie C++ care să fie adevărată dacă și numai dacă cele două cercuri $c1$ și $c2$, de centre diferite, nu se intersectează. (6p.)

3. În secvența următoare, a este un tablou bidimensional cu 5 linii și 5 coloane. Fără a utiliza alte variabile, completați secvența cu instrucțiunile care generează elementele tabloului următor:

```
for(i=0;i<5;i++)
for(j=0;j<5;j++)
.....
```

```
25 24 23 22 21
20 19 18 17 16
15 14 13 12 11
10 9 8 7 6
5 4 3 2 1
```

(6p.)

SUBIECTUL al III-lea**(30 de puncte)**

1. Subprogramul **comune** are doi parametri, a și b , prin care primește câte un număr natural ($a \in [0, 10^9]$, $b \in [0, 10^9]$). Subprogramul returnează cel mai mare număr care se poate forma cu cifrele distincte comune celor două valori, sau valoarea -1 în cazul în care cele două valori nu au nicio cifră comună. Scrieți definiția completă a subprogramului.

Exemplu: Dacă $a=806528$ și $b=207068$, subprogramul returnează numărul 8620 .

(10p.)

2. Se citește de la tastatură un șir de caractere, format din maximum 250 de litere mari și mici ale alfabetului englez. Cuvintele textului sunt separate între ele prin caracterele spațiu, '.' și ', '. Să se afișeze pe ecran toate cuvintele din textul citit care conțin același număr de vocale ca și ultimul cuvânt, sau mesajul **“nu există”**, dacă niciun cuvânt din text nu îndeplinește această condiție. Cuvintele vor fi scrise fiecare pe câte o linie de ecran.

Exemplu: Dacă se citește șirul **Scoala online va deveni un mod de instruire obisnuit pentru elevi.**, se vor afișa pe ecran:

Scoala
online
deveni

(10p.)

3. Fișierul **“bac.txt”** conține un șir de cel mult 10^9 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran în ordine **creștătoare**, separate cu spațiu, toate valorile de două cifre aflate în șirul citit din fișier, care apar de cele mai multe ori ca prefixe ale numerelor cu mai mult de două cifre aflate în fișier. Dacă nu există astfel de valori, se va afișa pe ecran mesajul **“nici o valoare”**.

Proiectați un algoritm eficient din punctul de vedere al timpului de execuție.

Exemplu: Dacă fișierul bac.txt conține valorile 1234 **78** **12** 978 132 128 7800 222 97 7831 13, se afișează pe ecran valorile 12 78.

a. Scrieți programul C/C++ corespunzător algoritmului proiectat.

(8p.)

b. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

(2p.)

REZOLVARE SUBIECTE
TEST 10
SUBIECTUL I
(20 de puncte)

1. c) 2. a) 3. c) 4. b) 5. b)

SUBIECTUL al II-lea
(40 de puncte)

1.

a) 5 180

b) Orice doua valori ce definesc extremitatile unui interval ce nu contine niciun multiplu de 17

c)

```

#include <iostream>
using namespace std;
int a,b,c,x,n,s;
int main()
{
    cin>>a>>b>>c;
    if(a>b)
    {
        a=a+b;
        b=a-b;
        a=a-b;
    }
    for(x=b;x>=a;x--)
        if(x%c==0)
        {
            n++;
            s=s+x;
        }
    cout<<n<<' '<<s;
    return 0;
}

```

d)

citește a, b, c (numere naturale nenule)

 ┌dacă $a > b$ atunci

 | $a \leftarrow a + b$

 | $b \leftarrow a - b$

 | $a \leftarrow a - b$

└─

 $n \leftarrow b/c - (a-1)/c$

 ┌dacă $a \% c \neq 0$ atunci

 | $a \leftarrow (a/c + 1) * c$

└─

 $b \leftarrow b/c * c$
 $s \leftarrow (a + b) * n / 2$

Scrie n, ' ', s

 2. $\sqrt{\text{pow}(c1.\text{centru}.x - c2.\text{centru}.x, 2) + \text{pow}(c1.\text{centru}.y - c2.\text{centru}.y, 2)} > c1.r + c2.r$

3. for(i=1; i<=5; i++)

for(j=1; j<=5; j++)

 $a[i][j] = 5 * (4 - i) + (5 - j);$

SUBIECTUL al III-lea

(30 de puncte)

1.

```
int comune(int a, int b)
{
    int va[10]={0},vb[10]={0},x=0,c,k=0;
    while(a)
    {
        va[a%10]++;
        a=a/10;
    }
    while(b)
    {
        vb[b%10]++;
        b=b/10;
    }
    for(c=9;c>=0;c--)
        if(va[c]&&vb[c])
        {
            x=x*10+c;
            k++;
        }
    if(k)
        return x;
    return -1;
}
```

2.

```
#include <bits/stdc++.h>
using namespace std;
char s[251],*p,v[200][25],voc[]="aeiouAEIOU";
int n, i, nvi,nvu,ok,j;
```

```
int main()
{
    cin.getline(s,251);
    p= strtok(s, " ,.");
    while(p)
    {
        strcpy(v[++n],p);
        p= strtok(NULL, " ,.");
    }
    for(i=0;i<strlen(v[n]);i++)
        if(strchr(voc,v[n][i]))
            nvu++;
    for(i=1;i<n;i++)
    {
        nvi=0;
        for(j=0;j<strlen(v[i]);j++)
            if(strchr(voc,v[i][j]))
                nvi++;
        if(nvi==nvu)
        {
            cout<<v[i]<<endl;
            ok=1;
        }
    }
}
```

```

}
if(!ok)
  cout<<"nu exista" ;
return 0;
}

```

3.a)

```

#include <bits/stdc++.h>
using namespace std;
ifstream fin("bac.txt");

int a[100],b[100],x,maxi, ok;

int main()
{
  while(fin>>x)
  {
    if(x>9&&x<100)
      a[x]++;
    else
      if(x>99)
      {
        while(x>99)
          x=x/10;
        b[x]++;
      }
  }
  for(x=10;x<100;x++)
    if(a[x]&&b[x])
      if(a[x]>maxi)
        maxi=a[x];
  for(x=10;x<100;x++)
    if(a[x]&&b[x])
      if(a[x]==maxi)
      {
        cout<<x<<' ';
        ok=1;
      }
  if(!ok)
    cout<<"nici o valoare";
  return 0;
}

```

b) Algoritmul citește secvențial valorile, contorizează frecvența numerelor de două cifre în vectorul a, iar prefixele de două cifre sunt contorizate în vectorul b; După determinarea maximului de apariții, se parcurg pozițional și se afișează valorile care îndeplinesc cerința enunțului. Algoritmul este eficient ca timp de execuție deoarece face determinările cerute odată cu citirea datelor, deci este liniar.

BAREM DE EVALUARE ȘI DE NOTARE**TEST 10**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL al III-lea**(20 de puncte)**

1.c) ; 2.a) ; 3. c) ; 4. b) ; 5. b)	5x4p
-------------------------------------	------

SUBIECTUL al II-lea**(40 de puncte)**

1	a) Răspuns corect: 5 180	6p.	Se acordă câte 3p. pentru fiecare dintre cele două valori conform cerinței.
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (orice două valori care nu conțin, în intervalul delimitat, niciun multiplu de 17).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie (*) -instrucțiune repetitivă -atribuiri -corectitudine globală a programului ¹⁾	10p 1 p. 1p. 1p 3p 2p 1p 1p	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile de decizie este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței -corectitudine globală a algoritmului ¹⁾	6p.	Se acordă 2p pentru determinarea numărului de multipli de c din intervalul [a,b] și 4p pentru calculul sumei progresiei aritmetice de rație c, termen inițial a (adus la valoarea primului multiplu de c din interval) și termen final b (adus la valoarea ultimului multiplu de c din interval)
2	Pentru rezolvare corectă -referire corectă la campurile structurii/înregistrării -utilizarea corectă a formulei pentru distanța dintre centre -corectitudine globală a secvenței ¹⁾	6p. 2p 3p 1p	Condiția cerută va verifica dacă distanța dintre centrele cercurilor este mai mare decât suma razelor
	3	Pentru rezolvare corectă -acces la un element al tabloului -atribuire corectă a valorilor, conform cerinței (*) -corectitudine globală a secvenței ¹⁾	6p. 1p 4p 1p

SUBIECTUL al III-lea**(30 de puncte)**

1.	Pentru subprogram corect -antet subprogram (*) -determinare a valorii cerute (**) - instrucțiune/instrucțiuni de returnare a rezultatului -declarare a tuturor variabilelor locale, corectitudine globală a subprogramului ¹⁾	10p . 2p. 6p. 1p. 1p	(*) Se acordă câte 1p. pentru fiecare aspect al antetului (structură, parametri de intrare) conform cerinței. (**) Se acordă câte 2p. pentru fiecare aspect al cerinței (determinare cifre comune, formarea valorii rezultat, tratarea cazului particular).
2.	Pentru program corect -declarare a unei variabile care să memoreze un sir de caractere si a unui vector de siruri -citirea datelor -separarea cuvintelor din text, cu memorarea lor(*) - identificarea cuvintelor care respectă cerinta (**) -afisarea rezultatelor(tratarea cazului “nu există”) -declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p . 1p. 1p. 2p. 4p. 1p 1p	(*) Se acordă câte 1p. pentru fiecare aspect specific (identificare unui cuvânt, memorarea prin copiere) (**) Determinarea numărului de vocale din ultimul cuvânt, determinarea numărului de vocale din fiecare cuvânt analizat.
3.	a) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului	2p. 1p. 1p 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar . O soluție posibilă citește valorile din fișier, numără valorile de două cifre într- un vector de frecvență , numără prefixele de două cifre ale numerelor ≥ 100 în alt vector de frecvență, parcurge apoi pozitional cei doi vectori, determinând numărul maxim de aparitii(dintre acele valori x care au fost numarate în ambii vectori) și marcând existenta unor astfel de valori. Afisarea se face prin parcurgerea si identificarea valorilor care au numărul de aparitii egal cu valoarea maximă determinată.



INFORMATICĂ

Filiera teoretică, profil real, specializarea științele naturii

MODEL TEST 1

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată).

Subiectul I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele **a, b** și **z** sunt întregi, iar **a ≤ b**. Care dintre expresiile C/C++ următoare are valoarea 1 dacă și numai dacă valoarea variabilei **z** este pară și nu aparține intervalului închis determinat de valorile variabilelor **a** și **b** ?

- a.** `z%2==0 && z>a || z>b` **b.** `!(z<a && z>b) && z%2==0`
c. `z<a && z>b && z%2==0` **d.** `!(z>=a && z<=b) && z%2==0`

2. Se consideră secvența de instrucțiuni următoare:

```
int n,i=1,k=1;
cin>>n;
while(k*k<=n)
{
    i++;
    k=k+i;
}
```

`cout<<k;`

Ce valoare se afișează dacă pentru **n** se citește valoarea 99.

- a. 0** **b. 20** **c. 5** **d. 10**

3. Care dintre următoarele variante de instrucțiuni inserează cifra 2 înaintea ultimei cifre a unui număr natural **n**.

1. `n=(n%10*10+2)*10+n/10;` **2.** `n=(n/10*10+2)*10+n%10;`

3. `n=n/10+2*10+n%10;` **4.** `n=(n/10+2)*10+n%10;`

- a. 1** **b. 2** **c. 2, 3** **d. 3, 4**

4. Care dintre expresiile următoare are valoarea 1 dacă și numai dacă valorile variabilelor **a** și **b** sunt numere întregi pare consecutive.

a. `(a%2)&&(b%2)&&(a-b==2)` **b.** `(a%2)&&(a-b==2||b-a==2)`

c. `(a%2==0)&&!(abs(a-b)==2)` **d.** `(a%2==0)&&(abs(a-b)==2)`

5. Pentru tabloul unidimensional (**4,6,14,25,61,73,82,87,95,96,98**) numărul minim de elemente ale tabloului care trebuie verificate până este găsit elementul **82** este:

- a. 7** **b. 2** **c. 3** **d. 4**

Subiectul II**(40 de puncte)****1. Se consideră algoritmul alăturat, descris în pseudocod.**

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întregă a numărului real z .

a) Scrieți valorile care se vor afișa dacă se citesc în ordine numerele 5 15 45 33 81 66 44 87. **(6p.)**

b) Dacă pentru n , a și b se citesc valorile 5 50 100 completați setul de date cu valori care pot fi citite astfel încât, în urma executării algoritmului, valoarea afișată pentru m să fie **4**. **(6p.)**

c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

d) Scrieți în pseudocod un algoritm echivalent cu cel dat în care structura **pentru ...execută**

să fie înlocuită cu o structură repetitivă cu test inițial. **(6p.)**

citeste n, a, b (numere naturale nenule)

$m \leftarrow 0$

┌ pentru $i \leftarrow 1, n$ execută

| citește x

| ┌ dacă $x \geq a$ și $x \leq b$ atunci

|| ┌ dacă $x \% 10 = [x/10]$ atunci

|| | ┌ $m \leftarrow m + 1$

|| | └

|| └

└

scrie m

2. Tabloul unidimensional A, cu 5 elemente având valori distincte, memorează cele mai mici 5 numere naturale nenule pătrate perfecte. Tabloul unidimensional B, cu 4 elemente având valori distincte, memorează cele mai mici 4 numere naturale prime. Tablourile A și B sunt sortate descrescător. Se interclasează descrescător cele două tablouri A și B în tabloul unidimensional C. Precizați care sunt elementele tabloului C. **(6p.)**

3. Variabilele întregi i și j memorează numere naturale. Precizați ce se afișează după executarea instrucțiunilor de mai jos. **(6p.)**

```
for(i=0; i<=3; i++)
    for(j=3; j>=i; j--)
        if (j%3==2)
            cout<<i+j;
```

Subiectul III**(30 de puncte)**

1. Un șir bicolor este reprezentat sub forma unui tablou unidimensional cu n elemente ce conține numere naturale din mulțimea $\{0,1\}$. spunem că șirul este **perfect** dacă există o singură secvență cu indici în intervalul $i1, i2$ cu toate elementele sale egale cu 0 iar șirul nu conține alte elemente egale cu 0 . Scrieți programul C/C++ care citește de la tastatură numerele naturale n ($2 \leq n \leq 200$) și apoi n valori din mulțimea $\{0,1\}$ (cel puțin 2 valori distincte) reprezentând elementele șirului și afișează pe ecran mesajul **DA**, dacă șirul este perfect sau mesajul **NU** în caz contrar. **(10p)**

Exemplu: pentru $n=5$ și tabloul alăturat, **1 0 0 0 1**, se va afișa **DA**

Exemplu: pentru $n=5$ și tabloul alăturat, **1 0 1 0 1**, se va afișa **NU**

2. Un număr natural nenul se numește **echilibrat** dacă suma cifrelor de pe poziții pare este egală cu suma cifrelor de pe poziții impare. Cifrele se numerotează de la dreapta la stânga începând cu valoarea 0 . Exemplu: **121** este număr echilibrat pentru că $2=1+1$. Scrieți un program în pseudocod care citește două numere naturale a și b ($2 \leq a < b \leq 10^9$, $b-a \leq 10000$) și afișează pe ecran, în ordine descrescătoare, separate prin câte un spațiu, toate numerele **echilibrate** din intervalul $[a,b]$. Dacă în interval nu există astfel de numere, se afișează pe ecran mesajul **nu exista**. **(10p)**

Exemplu: pentru $a=100$ $b=150$, se afișează pe ecran: **143 132 121 110**.

3. O secvență de K elemente a unui șir de numere naturale este numită secvență **RK**, dacă elementele din secvență dau resturi distincte la împărțirea cu K . Fișierul **bac.txt** conține pe prima linie un număr natural K din intervalul $[1,10]$, iar pe a doua linie conține un șir de cel puțin K și cel mult 10^3 numere naturale din intervalul $[0,10^4]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran suma elementelor primei secvențe **RK** din șirul de pe a doua linie a fișierului, dacă în șir există secvențe **RK** sau mesajul **NU EXISTA** dacă șirul de pe a doua linie a fișierului nu conține nicio secvență **RK**. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplul 1. Dacă fișierul conține numerele

3

10 10 11 3 4 2 49 21 27 12 13 atunci se afișează pe ecran 24

(secvența **RK** 11 3 4)

Exemplul 2. Dacă fișierul conține numerele

3

10 11 13 16 11 10 atunci se afișează pe ecran **NU EXISTĂ** (în șir nu există nicio secvență **RK**).

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)

b) Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p)

REZOLVARE SUBIECTE

TEST 1

Subiectul I

(20 de puncte)

1. d.!(z>=a && z<=b)&&z%2==0
2. d.10
3. b. 2
4. d.(a%2==0)&&(abs(a-b)==2)
5. c.3

Subiectul II

(40 de puncte)

1.a) 2 (6p.)

b) $n=5$ $a=50$ $b=100$ putem citi valorile 55 66 77 88 90 sau orice set format din 5 valori din care 4 sunt din mulțimea {55, 66, 77, 88, 99} (6p.)

c) #include <iostream>

using namespace std;

int main()

```

{
    int n, a, b, i, x, m;
    cin>>n>>a>>b;
    m=0;
    for(i=1;i<=n;i++)
    {
        cin>>x;
        if(x>=a&&x<=b)
            if(x%10==x/10)
                m++;
    }
    cout<<m;
    return 0;
}

```

d. structura repetitivă cu test initial este structura cat timp ... executa (6p.)

citeste n, a, b (numere naturale nenule)

 $m \leftarrow 0; i \leftarrow 1$ ┌cat timp $i \leq n$ execută

| citește x

| ┌dacă $x \geq a$ și $x \leq b$ atunci|| ┌dacă $x \% 10 = [x / 10]$ atunci|| | $m \leftarrow m + 1$

|| └─

| └─

| $i \leftarrow i + 1$

└─

scrie m

2. A=(25, 16, 9, 4, 1) B=(7, 5, 3, 2) C=(25, 16, 9, 7, 5, 4, 3, 2, 1). (6p.)

3. Se afișează 234 (6p.)

Subiectul III

(30 de puncte)

1. Vom memora tabloul și vom determina prima și ultima apariție a unui element egal cu 0. Verificăm dacă elementele cuprinse între cele două apariții sunt toate egale cu 0.

```
#include <iostream>
using namespace std;
int main()
{
    int n, v[201], i,i1=0,i2=0, ok=1,primul=0;
    cin>>n;
    for(i=1;i<=n;i++)
    {
        cin>>v[i];
        if(v[i]==0&&primul==0)
            {i1=i;primul=1;}
        if(v[i]==0)i2=i;
    }
    for(i=i1;i<=i2;i++)
        if(v[i]!=0)ok=0;
    if(primul==0||ok==0)cout<<"NU";
    else cout<<"DA";
    return 0;}

```

2. Căutăm numerele echilibrate numai printre numerele divizibile cu 11.

întreg a, b, ok,si,sp,ci,r;
citeste a, b (numere naturale nenule)
ok←0

```
┌pentru i←b,a,-1 execută
| ┌dacă i%11=0 atunci
|| ci←i; si←0; sp←0; r←0;
|| ┌cat timp ci>0 executa
||| ┌dacă r%2=0 atunci
||| | sp←sp+ci%10;
||| |altfel
||| | si←si+ci%10;
||| └─┘
||| r←r+1; ci←[ci/10]
|| └─┘
|| ┌dacă si=sp atunci
||| scrie i,' ';
||| ok←1
|| └─┘
└─┘

┌dacă ok=0 atunci
| scrie "nu exista"
└─┘

```

3. a) O soluție eficientă utilizează un vector cu k elemente cu ajutorul căruia vom memora secvența curentă cu k elemente prin utilizarea vectorului circular, pentru a vedea dacă resturile la împărțirea cu k sunt distincte folosim un vector de apariții, dar și contorizăm câte resturi distincte avem în secvența curentă cu ajutorul variabilei nr , iar cu variabila s determinăm suma secvenței curente. Algoritmul este eficient din punct de vedere al spațiului de memorie deoarece utilizează doar variabile simple și 2 vectori a câte 10 elemente ($k \leq 10$). Eficiența timp este dată de faptul că este un algoritm liniar, la o singură trecere prin fișier determină valoarea cerută, complexitatea algoritmului depinde doar de numărul de valori din fișier.
4. b)

```

#include <iostream>
#include <fstream>
using namespace std;
ifstream fin("bac.txt");
int k, i, x, v[10], s, ap[10], nr, ok, y;
int main()
{
    fin >> k;
    for(i=0; i<k; i++)
    {
        fin >> v[i];
        ap[v[i]%k]++;
        if(ap[v[i]%k]==1) nr++;
        s=s+v[i];
    }
    if(nr==k){ok=1; cout << s; return 0;}
    while(fin >> x)
    {
        y=v[i%k];
        s=s-y;
        if(ap[y%k]==1) nr--;
        ap[y%k]--;
        s=s+x; v[i%k]=x;
        ap[x%k]++;
        if(ap[x%k]==1) nr++;
        if(nr==k){cout << s; ok=1; return 0;}
        i++;
    }
    if(ok==0) cout << "NU EXISTA";
    return 0;
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 1**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

1. d	2. d	3. b	4. d	5. c	5x4p.
------	------	------	------	------	-------

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	6p.	orice set format din 5 valori din care 4 sunt din mulțimea {55, 66, 77, 88, 99}
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiunea repetitivă -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principal corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă A=(25, 16, 9, 4, 1) B=(7, 5, 3, 2) C=(25, 16, 9, 7, 5, 4, 3, 2, 1). (*)	6p. (**)	(*) Se acordă câte 2p. pentru fiecare aspect (determinarea corectă al fiecărui vector) (**) Se acordă întreg punctajul dacă vectorul final este corect, dar nu au fost precizați vectorii intermediari
3.	Pentru rezolvare corectă 234 (*)	6p. .	(*) Se acordă numai 3p. dacă doar o parte dintre valorile afișate sunt conform cerinței.

SUBIECTUL al III - lea**(30 de puncte)**

1.	Pentru program corect -declararea variabilelor -citire a datelor -verificarea proprietăților cerute, (*) -corectitudine a globală a programului ¹⁾	10p. 1p. 2p. 6p. 1p	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (valorile egale cu 0 în același interval, valorile egale cu 1 în afara intervalului, cazul cand nu sunt îndeplinite condițiile din enunț).
2.	Pentru algoritm corect - citire date, inițializare (*) - determinare valorilor cerute (**) - verificarea existenței valorilor cerute -corectitudine globală a programului ¹⁾	10p. 2p. 6p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect conform cerinței. (**) Se acordă câte 3p. pentru fiecare aspect al cerinței (calculul celor două sume, condiția de afișare).
3.	a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinarea valorilor cerute (*),(**) -utilizarea unui algoritm eficient (***) -declarare a variabilelor, afișare a datelor -corectitudine globală a programului ¹⁾	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar. O soluție eficientă utilizează un vector cu k elemente cu ajutorul căruia vom memora secvența curentă cu k elemente prin utilizarea vectorului circular, pentru a vedea dacă resturile la împărțirea cu k sunt distincte folosim un vector de apariții, dar și contorizăm câte resturi distincte avem în secvența curentă cu ajutorul variabile nr, iar cu variabila s determinăm suma secvenței curente. Algoritmul este eficient din punct de vedere a spațiului de memorie deoarece utilizează doar variabile simple și 2 vectori a câte 10 elemente ($k \leq 10$). Eficiența timp este data de faptul că este un algoritm liniar, la o singura trecere prin fișier determină valoarea cerută, complexitatea algoritmului depinde doar de numărul de valori din fișier.

MODEL TEST 2

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Stabiliți care dintre expresiile următoare are valoarea 1 dacă și numai dacă variabila reală x aparține reuniunii intervalelor $[2,10]$ și $(30,40)$?

- a) $!((x < 2 \parallel x > 10) \parallel (x \geq 40 \parallel x \leq 30))$
 b) $(x \leq 10 \ \&\& \ x \geq 2) \ \&\& \ (x > 30 \ \&\& \ x < 40)$
 c) $(x > 10 \ \&\& \ x \geq 2) \ \&\& \ (x \leq 30 \ \&\& \ x \leq 40)$
 d) $!(x < 2 \parallel x > 10) \parallel !(x \geq 40 \parallel x \leq 30)$

2. Stabiliți ce valoare au variabilele x , y și z ce memorează numere naturale astfel încât expresia C++

$10/x/y+z\%5*x/3$ să aibă valoarea 200.

- a) $x=100,y=2,z=23$ b) $x=200,y=1,z=23$ c) $x=200,y=2,z=20$ d) $x=150,y=5,z=8$

3. Se consideră variabilele i, j și nr de tip întreg. Ce elemente va avea tabloul a , după execuția secvenței de program C/C++ următoare?

```
nr=0;
for(i=2;i<=3;i++)
for(j=2;j<=4;j++)
{
  if(i%j>j%i) a[nr]= i%j;
  else   a[nr]= j%i;
  nr++;
}
```

- a) (0,1,0,1,0,1) b) (0,2,0,2,0,2) c) (0,2,2,2,0,3) d) (1,1,2,1,1,1)

4. Se consideră tabloul unidimensional a , care conține, în ordine crescătoare, cele mai mici 5 numere prime de 2 cifre și tabloul unidimensional b care conține, în ordine crescătoare, cele mai mici 4 numere naturale de două cifre, care au suma cifrelor un număr prim. Prin interclasarea vectorilor a și b se va obține tabloul:

- a) (11,12,13,14,16,17,18,19,23) b) (11,11,12,13,14,15,19,19,23)
 c) (11,11,12,13,13,16,17,19,23) d) (11,11,12,13,14,16,17,19,23)

5. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

Ce se va afișa prin execuția algoritmului, dacă $a=31287$ și $b=1887$?

- a. 187 b. 87 c. 3 d.4

citește a,b (a,b numere naturale nenule)

$y \leftarrow 0$

```

- repeta
  c1 ← a%10; c2 ← b%10
  dacă c1=c2 atunci
    y ← y+1
  a ← [a/10]
  b ← [b/10]
- până când a=0 sau b=0
scrie y

```

SUBIECTUL al II-lea
(40 puncte)
1. Se consideră algoritmul alăturat descris în pseudocod:

S-a notat cu $x \% y$ restul împărțirii numerelor întregi x și y și cu $[x]$ partea întreagă a numărului real x .

- a) Ce se va afișa, dacă se citesc, în această ordine, numerele **5,3,2324,31,7229,361,75** (6p.).
- b) Scrieți un set de date de intrare care să determine afișarea valorii **200**. (6p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat, în care să se înlocuiască structura **pentru** cu alt tip de structură repetitivă. (6p.)

citește n,c (n, c numere naturale nenule, $c < 10$)

$e \leftarrow 0$

```

- pentru i ← 1, n execută
  citește x (numar natural nenul)
  p ← 1
  nr ← 0
  cx ← x
  cat timp x ≠ 0 executa
    dacă x%10 ≤ c atunci
      nr ← nr + (x%10)*p
      p ← p*10
    x ← [x/10]
  dacă nr%2=0 și nr > 0 atunci
    e ← e + cx
scrie e

```

2. Se consideră declarațiile de mai jos, în care variabilele **numarator1** și **numitor1** memorează două numere naturale nenule ce reprezintă numărătorul și numitorul primei fracții iar **numarator2** și **numitor2** memorează numărătorul și numitorul celei de a doua fracții.

int numarator1,numitor1,numarator2,numitor2,numarator_suma,numitor_suma;

Scrieți o secvență de instrucțiuni C/C++ care determină suma celor două fracții, memorând în variabilele **numarator_suma** și **numitor_suma** numărătorul și numitorul sumei celor două fracții. (6p.)

3. În secvența de instrucțiuni de mai jos, variabilele **i** și **j** sunt de tip întreg. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze caracterele de mai jos, în această ordine. (6p.)

```
for (i=0; i<5; i++) {
    for (j=0; j<5; j++)
        .....
}
```

0	1	2	3	4
1	2	3	4	0
2	3	4	0	1
3	4	0	1	2
4	0	1	2	3

SUBIECTUL al III - lea**(30 puncte)**

1. Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ($1 \leq n \leq 100$) și un tablou unidimensional **v**, format din **n** numere naturale nenule și distincte, fiecare număr având cel mult 9 cifre și afișează numărul de perechi distincte formate din elemente din tabloul **v**, care sunt prime între ele. Două numere naturale sunt prime între ele dacă au un singur divizor comun: numărul 1. Două perechi de elemente sunt distincte dacă diferă prin cel puțin un element. **Exemplu:** dacă **n=6** și **v=(4,22,12,9,63,28)** atunci se va afișa valoarea **5**, deoarece vectorul are **5** perechi de numere ce verifică proprietatea dată: **(4,9);(4,63);(22,9);(22,63);(9,28)**. (10p.)

2. Se citește de la tastatură un număr natural **n** de cel mult 9 cifre, care are cel puțin o cifră impară. Scrieți un program în limbaj pseudocod care determină numărul **m** obținut prin eliminarea cifrelor pare ale numărului **n** și afișează oglinditul numărului **m**. **Exemplu:** dacă **n=5768132**, se va obține **m=5713** și se va afișa oglinditul său egal cu **3175**. (10p.)

3. Fișierul BAC.TXT conține pe prima linie două numere naturale nenule **n** și **m** ($n \leq 1000$, $m \leq 10000$). A doua linie din fișier conține un șir **a** având **n** numere naturale, formate din cel mult trei cifre, în ordine strict crescătoare și separate prin câte un spațiu. A treia linie din fișier conține un șir **b** având **m** numere naturale, formate din cel mult trei cifre și separate prin câte un spațiu. Scrieți un program C/C++ care citește valorile din fișierul BAC.TXT și afișează pe ecran, în ordine crescătoare și separate prin spațiu, numerele din șirul **b** ce ar putea fi inserate în șirul **a**, astfel încât **a** să rămână ordonat strict crescător sau afișează mesajul **Nu exista numere care vor fi inserate**, dacă niciun număr din șirul **b** nu poate fi inserat în șirul **a** cu proprietatea de mai sus. Se va utiliza un algoritm eficient din punct de vedere al memoriei utilizate și al timpului de execuție. De exemplu, dacă fișierul BAC.TXT are conținutul următor, pe ecran se va afișa **3 35 90**.

6 7
5 14 25 80 100 150
5 3 90 14 35 3 35

a. Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

(2p.)b. Scrieți programul C/C++ corespunzător algoritmului descris la punctul a.

(8p.)

REZOLVARE SUBIECTE
TEST 2
Subiectul I

1. d 2. b 3. c 4. d 5. c

Subiectul al II-lea

1.

a) 9553

b) Algoritmul adună dintre cele n numerele citite pe cele care respectă condiția: numărul obținut prin eliminarea cifrelor strict mai mari decât c este par. Un set de numere posibile: 5, 6, 34, 128, 12, 26, 179, deoarece se va obține suma egală cu $34+128+12+26=200$

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    unsigned int n,c,e,i,x,p,nr,cx;
    cin>>n>>c;
    e=0;
    for(i=1;i<=n;i++){
        cin>>x;
        p=1;
        nr=0;
        cx=x;
        while(x!=0){
            if(x%10<=c){
                nr=nr+(x%10)*p;
                p=p*10;
            }
            x=x/10;
        }
        if(nr%2==0 && nr>0)
            e=e+cx;
    }
    cout<<e;
    return 0;
}
```

d) Programul în limbaj pseudocod echivalent este:

```
citește n,c (n, c numere naturale nenule, c<10)
e←0
i←1
cât timp i≤n execută
    p←1
    nr←0
    cx←x
    cât timp x≠0 execută
        dacă x%10≤c atunci
            nr←nr+(x%10)*p
            p←p*10
        x←[x/10]
    dacă nr%2=0 și nr>0 atunci
        e←e+cx
    i←i+1
scrie e
```

2. numarator_suma=numarator1*numitor2+numarator2*numitor1;
numitor_suma=numitor1*numitor2;
3. cout<<(i+j)%5<<' '; cout<<endl;

Subiectul al III-lea

1.

```

#include<iostream>
using namespace std;
int main(){
    int a,b,r,i,j,v[101],n,nr;
    cout<<"n=";
    cin>>n;
    for(i=1;i<=n;i++){
        cout<<"v["<<i<<"]="";
        cin>>v[i];
    }
    nr=0;
    for(i=1;i<n;i++){
        for(j=i+1;j<=n;j++){
            a=v[i];
            b=v[j];
            while(b!=0){
                r=a%b;
                a=b;
                b=r;
            }
            if(a==1)
                nr++;
        }
    }
    cout<<"Numarul de perechi este: "<<nr;
    return 0;
}

```

2. citește n (n număr natural)

 $m \leftarrow 0$ $p \leftarrow 1$

cât timp (n ≠ 0) execută

dacă (n%2=1) atunci

 $m \leftarrow n \% 10 * p + m$ $p \leftarrow p * 10$ $n \leftarrow [n/10]$ $r \leftarrow 0$

cât timp (m ≠ 0) execută

 $r \leftarrow r * 10 + m \% 10$ $m \leftarrow [m/10]$

scrie r

3. a. Programul utilizează un vector v cu 1000 de elemente, corespunzător valorilor posibile ale numerelor citite, în care se completează valoarea 1 pentru numerele din șirul a simultan cu citirea lor. La citirea fiecărui număr x din șirul b se verifică dacă $v[x]=0$ (ce semnifică faptul că nu există valoarea x în șirul a și poate fi inserată) și se completează cu valoarea 2. Algoritmul este eficient ca spațiu de memorie deoarece utilizează un singur vector și este eficient ca timp de execuție deoarece parcurge o singură dată fiecare șir de numere (algoritm liniar).

b.

```
#include<iostream>
#include<fstream>
using namespace std;
int v[1000];
int main(){
    ifstream fin("BAC.TXT");
    int n,m,i,a,b,ok;
    ok=0;
    fin>>n>>m;
    for(i=1;i<=n;i++){
        fin>>a;
        v[a]=1;
    }
    for(i=1;i<=m;i++){
        fin>>b;
        if(v[b]==0){
            v[b]=2;
            ok=1;
        }
    }
    if(ok==0)
        cout<<"Nu exista numere care vor fi inserate";
    else{
        for(i=0;i<=999;i++)
            if(v[i]==2)
                cout<<i<<' ';
    }
    return 0;}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 2**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

1. d	2. b	3. c	4. d	5. c	5x4p.
------	------	------	------	------	-------

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 9553	6p.	
	b) Pentru răspuns corect	6p.	
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ²⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(* Se acordă numai 2p. dacă numai una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ²⁾	6p. 5p. 1p.	(* Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -determinarea numitorului sumei fracțiilor -determinarea numărătorului sumei fracțiilor -corectitudine globală a secvenței ¹⁾	6p. 3p. 2p. 1p.	
3.	Pentru rezolvare corectă -atribuire a valorilor indicate pentru afișare (*) -afișarea valorilor pe linii, conform cerinței	6p. 4p. 2p.	(* Se acordă numai 2p. dacă doar o parte dintre elementele afișate sunt conform cerinței.

¹ Corectitudinea globală vizează structura, sintaxa și alte greșeli neprecizate în barem.

SUBIECTUL al III - lea**(30 de puncte)**

<p>Pentru program corect -declararea corectă a variabilei de tip tablou</p> <p>1. -citire corectă a datelor de intrare - algoritm de verificare dacă două numere sunt prime între ele - determinarea numărului de perechi cerut -corectitudine globală a programului ²⁾</p>	<p>10p. 1p. 1p. 4p. 3p. 1p.</p>	
<p>Pentru program corect -declararea de variabile simple -citirea datelor -determinarea numărului m obținut după eliminarea cifrelor pare -accesul la cifrele numărului n -determinarea oglinditului numărului m -corectitudine a globală a programului ²⁾</p>	<p>10p. 1p. 1p. 2p. 2p. 3p. 1p.</p>	
<p>a) Pentru răspuns corect -coerența descrierii algoritmului (*) -justificare a unor elemente de eficiență</p> <hr/> <p>b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) 3. -utilizarea unui algoritm eficient (***) -declararea variabilelor, afișarea datelor, corectitudinea globală a programului ²⁾</p>	<p>2p. 1p. 1p.</p> <hr/> <p>8p. 1p. 5p. 1p. 1p.</p>	<p>.(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.</p> <p>(**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare.</p> <p>(***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă construiește, pe măsura citirii datelor, un vector caracteristic pentru elementele din șirul a și le notează cu 1. Apoi citește elementele șirului b și verifică pentru fiecare număr, pe baza vectorului caracteristic, dacă nu există un termen egal cu el din șirul a, în caz afirmativ acesta se notează cu 2 în vector. Se parcurge vectorul caracteristic, în sens crescător și se afișează numerele notate cu 2.</p>

MODEL TEST 3

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Care dintre următoarele expresii are valoarea 1, dacă **a** este o variabilă întreagă cu valoarea 2020?

a. $(a \% 3 / 10 == a / 10 \% 3) \&\& (a \% 3)$

b. $(a \% 2 / 10 == a / 100 \% 2) \&\& (a \% 11)$

c. $(a == a - 1) \&\& (a \% 7)$

d. $(a / 20! = 101) \&\& (a \% 5)$

2. Se consideră următorul șir de numere: 2, 4, 5, 7, 7, 7, 8. Aplicând metoda căutării binare, după al câtelea număr verificat se găsește în șir valoarea 2?

a. 1

b. 2

c. 3

d. 4

3. Se consideră următorul șir de numere: 12, 4, 25, 7, 17, 67, 8. Aplicând metoda sortării prin selecția minimului pentru sortarea crescătoare a șirului, câte interschimbări se vor efectua?

a. 4

b. 16

c. 3

d. 10

4. Care dintre următoarele variante realizează rotunjirea numărului real **x**, ce reprezintă media unui elev, la maxim două zecimale.

a. $\text{floor}(x)$

b. $\text{ceil}(x)$

c. $\text{int}(x+0.5)$

d. $\text{int}(x*100+0.5)/100.0$

5. Se dau următoarele două șiruri de numere:

a=(8,6,5,4,3,1)

b=(9,8,7,6,5,4,3)

Dacă se aplică metoda interclasării pe aceste șiruri, care este șirul obținut, știind faptul că se vor lua în considerare doar elementele comune care sunt divizibile cu 3.

a. 8, 6, 5, 4, 3

b. 9, 6, 3

c. 3, 6

d. 6, 3

SUBIECTUL al II-lea**(40 de puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $a\%b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[x]$ partea întregă a lui x .

a. Dacă pentru a se citește valoarea 14, iar pentru b valoarea 93 ce va afișa algoritmul? **(4 p)**

b. Dacă pentru a se citește valoarea 5, iar algoritmul afișează $172/33$, atunci ce valoare trebuie să aibă b ? **(6 p)**

c. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind structurile cât timp... execută cu structuri de alt tip. **(4 p)**

d. Scrieți programul C/C++ corespunzător algoritmului dat. **(6 p)**

```

citește a,b (b≠0)
f2←0
x←b
cât timp x≠0 execută
  f2←f2*10+9
  x←[x/10]
  ■
f1←a*f2+b
a←f1
b←f2
cât timp b≠0 execută
  r←a%b
  a←b
  b←r
  ■
scrie f1/a,'/',f2/a

```

2. Pentru o piesă de teatru se cunosc: prețul unui bilet, numărul de rânduri din sală, precum și numărul de locuri de pe un rând. Scrieți în limbajul C/C++ o secvență de instrucțiuni în care să declarați sugestiv variabilele necesare calculării sumei totale încasată pentru un spectacol la care s-au vândut toate biletele. Citiți valorile și afișați suma totală. **(10 p)**

3. Cu ce trebuie înlocuite punctele de suspensie din codul următor astfel încât să se afișeze cel mai mare număr natural par memorat în variabila a ? **(10 p)**

```

int a[]={45,2,6,13,4,5},i,p=-1;
for(i=0;i<6;i++)
  ...
cout<<v[p];

```

SUBIECTUL al III-lea**(30 de puncte)**

1. Să se scrie un algoritm în pseudocod care citește un șir de n numere naturale și care afișează cea mai mare fracție care se poate forma cu două dintre numerele citite.

De exemplu, dacă $n=5$ și cele 5 numere sunt 11, 45, 123, 544 și 23, atunci se va afișa $544/11$.

(10 p)

2. Să se scrie un program care citește o valoare naturală n (<1000) și un șir de n numere care conține toate valorile de la 1 la $n-1$, astfel că o singură valoare se repetă. Programul va afișa valoarea care se repetă.

De exemplu, dacă $a=(4,2,3,1,2)$ și $n=5$, atunci se va afișa valoarea 2.

(10 p)

3. Fișierul **info.txt** conține pe prima linie o valoare naturală n ($<10^5$), iar pe următoarea linie un șir de n numere naturale ($<10^9$) separate prin spații.

a. Scrieți un program care să citească din fișier șirul de numere și care determină eficient din punct de vedere al timpului de executare și al memoriei, câte dintre perechile de elemente din șir sunt formate din valori cu aceeași sumă a cifrelor. **(8 p)**

b. Descrieți succint, în limbaj natural, metoda de rezolvare folosită, explicând în ce constă eficiența ei. **(2 p)**

REZOLVARE SUBIECTE
TEST 3
SUBIECTUL I

1. b

2. c

3. a

4. d

5. d

SUBIECTUL al II – lea

1.

a. 493/33

b. 21

c.

```

citește a,b (b≠0)
f2←0
x←b
┌repetă
│ f2←f2*10+9
│ x←[x/10]
│ ── până când x=0
f1←a*f2+b
a←f1
b←f2
┌repetă
│ r←a%b
│ a←b
│ b←r
│ ── până când b=0
scrie f1/a,',f2/a

```

d.

```

#include<iostream>
using namespace std;
int a,b,f1,f2,r,x;
int main()
{
    cin>>a>>b;
    x=b;
    while(x)
        f2=f2*10+9, x=x/10;
    f1=a*f2+b;
    a=f1;
    b=f2;
    while(b)
        r=a%b, a=b, b=r;
    cout<<f1/a<<'/'<<f2/a;
}

```

2.

```

int pret, nr_locuri, nr_randuri,suma_totala;
cin>>pret>>nr_randuri>>nr_locuri;
suma_totala=pret*nr_randuri*nr_locuri;
cout<<suma_totala;

```

3.

```

if(a[i]%2==0)
    if(p==-1)
        p=i;
    else
        if(a[p]<a[i])
            p=i;

```


SUBIECTUL al III – lea

1.

```

citește n,a
mini←a
maxi←a
┌ pentru i←2,n execută
│ citește a
│ ┌ dacă mini>a atunci
│ │ mini←a
│ │ ──┘
│ └ dacă maxi<a atunci
│ │ maxi←a
│ │ ──┘
│ ──┘
scrie maxi, '/', mini

```

2.

```

#include<iostream>
using namespace std;
int a[1001], n,s,i
int main()
{
  cin>>n;
  for(i=0;i<n;i++)
    cin>>a[i];
  for(i=0;i<n;i++)
    s=s+a[i];
  cout<<s-n*(n-1)/2;
}

```

3.

a.

```

#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("info.txt");
int v[81],n,i,k,s;
int main()
{
  fin>>n;
  while(n--)
  {

```

```

    fin>>i;
    s=0;
    while(i)
      s=s+i%10,i=i/10;
    v[s]++;
  }
  for(i=0;i<=81;i++)
    k=k+v[i]*(v[i]-1)/2;
  cout<<k;
}

```

b.

Se utilizează un vector v în care $v[i]$ = câte numere citite au suma cifrelor i . Algoritmul este eficient din punct de vedere al timpului de executare deoarece se parcurg o singură dată numerele din fișier și este eficient din punct de vedere al memoriei deoarece se folosesc variabile simple și un vector de maxim 81 de elemente.

BAREM DE EVALUARE ȘI DE NOTARE**TEST 3**

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.

SUBIECTUL I**20 de puncte**

1.	b	4p	
2.	c	4p	
3.	a	4p	
4.	d	4p	
5.	d	4p	

SUBIECTUL II**40 de puncte**

1.	a	493/33	4p	
	b	21	6p	
	c	Pentru rezolvare corectă	4p.	Se acordă numai 2 puncte dacă doar una dintre cele două instrucțiuni este corectă
	d	Pentru program corect -declararea corectă a tuturor variabilelor -citire corectă -scriere corectă -instrucțiune repetitivă corectă -atribuiri corecte -corectitudinea globală a programului.	6p. 1p. 1p. 1p. 1p. 1p. 1p	
2.		Pentru rezolvare corectă -declarare corectă a variabilelor -citire a datelor -realizarea calculului -afișarea rezultatului -corectitudine globală	10p 3p 2p 2p 2p 1p	
3.		Pentru rezolvare corectă -instrucțiune decizională corectă(*) -instrucțiune de atribuire corectă(**)	10p 5p 5p.	(*) se acordă 1p dacă există instrucțiunea decizională, dar nu produce rezultatul așteptat (**)se acordă 1p dacă există instrucțiunea de atribuire, dar nu produce rezultatul așteptat

SUBIECTUL III**30 de puncte**

1.	Pentru algoritm corect - citire a datelor - determinare a numărului cerut (*) - scriere a datelor - scriere principial corectă a structurilor de control (**)	10p. 1p 6p 2p 1p	(*)Se acordă numai 2p. dacă algoritmul este principial corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare. (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
2.	Pentru subprogram corect -declarare variabile -citirea datelor -identificarea dublurii(*) -afișarea valorii cerute -corectitudine globală	10p. 1p 1p 5p 2p 1p	(*)Se acordă numai 2p. dacă algoritmul este principial corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare.
3.	a. Pentru program corect --operații cu fișiere: declarare, citire/ scriere din/în fișier -utilizare a unui algoritm eficient (**) -determinare și afișare a rezultatului conform cerinței,(***) -declarare variabile, corectitudine globală a programului	8p. 1p 2p 4p 1p	(**) Se acordă punctajul numai pentru un algoritm (de complexitate $O(n)$), care utilizează eficient memoria. (***) Se acordă numai 2p. dacă algoritmul este principial corect, dar nu conduce la rezultatul cerut pentru orice set de date de intrare.
	b. Pentru răspuns corect -coerența explicării metodei (*) -justificare a unor elemente de eficiență	2p. 1p 1p	(*) Se acordă punctajul chiar dacă metoda aleasă nu este eficientă.

Corectitudinea globală vizează structura, sintaxa si alte greșeli neprecizate în barem.

Subiectul II**(40 de puncte)**

1. În algoritmul alăturat reprezentat în pseudocod, s-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întregă a numărului real x .

- Scrieți valoarea afișată dacă se citesc valorile 51467 și 67. **(6p)**
- Dacă pentru b se citește valoarea 73, scrieți numărul valorilor de 4 cifre ce pot fi citite pentru a astfel încât valoarea afișată să fie divizibilă cu 50. **(6p)**
- Scrieți un algoritm echivalent cu cel dat care să utilizeze o structură repetitivă cu test final. **(6p)**
- Scrieți programul C/C++ corespunzător algoritmului dat **(10p)**

citește a, b (numere naturale) $p \leftarrow 1$ cât timp $a * b > 0$ și $a \% 10 = b \% 10$ execută $a \leftarrow [a/10]$ $b \leftarrow [b/10]$ $p \leftarrow p * 10$ $a \leftarrow a * p$ scrie a

2. O firmă de agenți de vânzări oferă un bonus de 5% (din vânzările efectuate de agent) dacă vânzările depășesc valoarea de X lei pe lună, sau 10% dacă vânzările acestuia depășesc Y lei pe lună. Scrieți instrucțiunile prin care se calculează în variabila reală b valoarea bonusului, știind că vânzările totale ale unui agent sunt memorate în variabila reală v .

(6p)

3. Radu deține un restaurant, iar ca să mărească vânzările, a decis să efectueze și livrări la domiciliu; la sfârșitul zilei, vrea să afle care a fost valoarea medie a unei comenzi livrate la domiciliu. Știind că numărul total de comenzi livrate la domiciliu este memorat în variabila de tip întreg n , valoarea totală a vânzărilor din acea zi este memorată în variabila t , iar valoarea totală a comenzilor livrate la domiciliu este memorată în variabila v , declarați cele 3 variabile și scrieți instrucțiunea care calculează și afișează valoarea medie a comenzilor livrate la domiciliu, rotunjită la întregul de sus. Variabilele t și v sunt de tip real.

(6p)**Subiectul al III – lea****(30 de puncte)**

- Scrieți un program C/C++ care citește de la tastatură a și b două numere naturale cu cel mult 9 cifre fiecare și afișează pe ecran cel mai mare număr cu cifrele impare distincte care apar atât în a cât și în b sau 0 dacă nu există astfel de cifre.

Exemplu: dacă $a=45013$ și $b=215578$ atunci se va afișa valoarea 51.

(10 p)

- Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul $[0, 10^9]$, în această ordine: numerele m , p și n , apoi cele n elemente ale unui tablou unidimensional de maxim 100 elemente. Programul modifică ordinea valorilor din tablou, astfel încât ultimele m poziții să fie ocupate de mulțimea formată din elemente ale tabloului divizibile cu p , iar primele $n-m$ poziții să fie ocupate de mulțimea celorlalte numere, într-o ordine oarecare.

Valorile tabloului vor fi afișate pe ecran, separate prin câte un spațiu. Dacă tabloul nu conține cel puțin m valori divizibile cu p , următoarea linie va afișa numărul de valori divizibile cu p , găsite.

Exemplul 1: $m=3$, $p=2$, $n=12$ și tabloul (21, 5, 18, 3, 7, 4, 10, 55, 40, 22, 19, 33) unul din șirurile afișate poate fi: (21, 5, 33, 3, 7, 19, 22, 55, 40, 10, 4, 18)

Exemplul 2: $m=5$, $p=5$, $n=12$ și tabloul (21, 5, 18, 3, 7, 4, 10, 55, 40, 22, 19, 33) unul din șirurile afișate poate fi (21, 33, 18, 3, 7, 4, 19, 22, 40, 55, 10, 5) apoi, pe rândul următor, valoarea 4.

(10 p)

3. Fișierul **BAC.IN** conține cel mult un milion de numere naturale de cel mult 9 cifre fiecare separate prin câte un spațiu. Să se afișeze pe ecran cel mai mare număr **a** de exact 3 cifre care nu apare în fișier pentru care există un număr **b**, care apare în fișier, astfel încât $a+b=1000$.

Dacă nu există niciun astfel de număr se va afișa mesajul **NU EXISTA**.

Proiectați un algoritm eficient din punct de vedere al timpului de executare și justificați eficiența acestuia.

Exemple:

BAC.IN	22 8425 4567 6 8 999 33 995 800 200 45123	8425 994 4567 5 6 994 3300 995 800 200 45123
Pe ecran	994	NU EXISTA

(10 p)

REZOLVARE SUBIECTE
TEST 4

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I

1. b	2. d	3. c	4. d	5. c
------	------	------	------	------

Subiectul II

1. a. 51400

b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$, $x \in \{1,2,\dots,9\}$, $y \in \{0,1,\dots,9\}$)

c. citește a,b (numere naturale)
 p←1
 dacă a*b>0 și a%10=b%10 atunci
 -repetă
 a←[a/10]
 b←[b/10]
 p←p*10
 -până când a*b=0 sau a%10≠b%10
 a←a*p
 scrie a

d.

```
#include<iostream>
using namespace std;
int main()
{ int a,b,p=1;
cin>>a>>b;
while (a*b>0 && a%10==b%10)
{ a=a/10;
  b=b/10;
  p=p*10;
}
a=a*p;
cout<<a;
return 0;
}
```

2. float b=0;
 if (v>y)
 b=v*0.10;
 else if (v>x)
 b=v*0.05;
3. int n;
 float t, v;
 cout<<ceil(v/n);

Subiectul III

1. #include<iostream>
 using namespace std;
 int main()
 { int i, ok1, ok2; long a, a1, b, b1, c=0;
 cin>>a>>b;
 for (i=9;i>=1;i-=2)
 { ok1=ok2=0; a1=a; b1=b;
 while (a1>0&&ok1==0) if (a1%10==i)ok1=1;

```

        else a1/=10;
        while (b1>0&&ok2==0)    if (b1%10==i) ok2=1;
                                else b1/=10;
        if (ok1==1 && ok2==1) c=c*10+i;
    }
    cout<<c;
    return 0;
}

2. #include <iostream>
using namespace std;
int main()
{
    int n, m, p, aux, i, j, t[200],jp;
    cin>>m>>p>>n;
    for (int i=0;i<n;i++)
        cin>>t[i];
    j=0; jp=n;
    for (i=0; i<n; i++)
    {
//cele divizibile le punem la final in afara sirului si apoi
//lipim subsirurile
        if (t[i]%p!=0) t[j++]=t[i];
        else t[jp++]=t[i];
    }
    i=j;
    j=n;
    while(i<n)
        t[i++]=t[j++];
    for (i=0;i<n;i++)
    {
        cout<<t[i]<<" ";
    }
    cout<<endl;
    if (jp-n < m)
    {
        cout<<jp-n<<endl;
    }
}

3. #include<iostream>
#include<fstream>
using namespace std;
int v[1000];
ifstream f ("BAC.IN");
int main()
{long x, ok=0;
while (f>>x) if (x<1000)v[x]++;
x=999;
while (x>=100&&ok==0)
    if (v[x]==0 && v[1000-x]!=0) ok=1;
    else x--;
if (ok) cout<<x;
else cout<<"NU EXISTA";
return 0;
}

```

O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifica primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Complexitatea algoritmului este liniară ($O(n)$).

BAREM DE EVALUARE ȘI DE NOTARE

TEST 4

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.

Subiectul I

20 de puncte

1. b	2. d	3. c	4. d	5. c	4p×5
------	------	------	------	------	------

Subiectul II

40 de puncte

4. a. 51400 b. 450 de valori (toate numerele de forma $\overline{xy73}$, $\overline{xy03}$, $\overline{xy53}$, $\overline{xy50}$, $\overline{xy00}$) c. echivalare corectă d. Pentru program corect _declararea corectă a tuturor variabilelor _citire corectă _scriere corectă _instrucțiune repetitivă corectă _atribuiri corecte corectitudinea globală a programului	6p 6p 6p 10p 2p 1p 1p 3p 2p 1p
5. Pentru rezolvarea corectă _declararea corectă a variabilei b _instrucțiunile de test corecte _formula bonusului corectă corectitudine generală	6p 1p 2p 2p 1p
6. Pentru rezolvare corectă _declararea corectă a variabilelor _instrucțiunea de scriere corectă formula de scriere corectă	6p 2p 2p 2p

Subiectul III

30 de puncte

1. Pentru rezolvare corectă _declarări corecte de variabile _citirea corectă a datelor de intrare _determinare cifre distincte comune _determinare număr maxim cerut _afișare corectă corectitudinea globală a programului	10 p 1p 1p 3p 3p 1p 1p
2. Pentru rezolvare corectă _declararea corectă a variabilelor _citirea corectă a datelor de intrare _algoritm principal corect _corectitudinea formală (variabile, structura, sintaxa) afișarea corectă a datelor de ieșire	10p 1p 2p 3p 2p 2p

3. Pentru rezolvare corectă	10p
_ operații cu fișiere (declarare, deschidere)	1p
_ citirea numerelor	1p
_ algoritm principal corect	2p
_ determinarea valorii cerute	2p
_ afișarea rezultatului	1p
_ corectitudine formală (declarare variabile, structură program, sintaxa instrucțiunilor)	1p
_ coerența explicării metodei	1p
_ explicarea unor elemente de eficiență conform cerinței (numai pentru eficiență $O(n)$)	1p
O soluție posibilă folosește un vector de frecvență pentru numerele de maxim 3 cifre care apar în fișier. Apoi se parcurge vectorul de la ultima poziție și se identifică primul număr (poziție) care are frecvența nulă și pentru care se îndeplinește condiția că $v[1000-nr]$ este nenulă. Astfel, algoritmul utilizat este de tip liniar (complexitate $O(n)$).	

MODEL TEST 5

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns se notează cu 4 puncte.

1. Variabila **n** memorează un număr natural cu 5 cifre. Care dintre expresiile C/C++ de mai jos schimbă cifra din mijloc cu cifra 0?

- | | |
|--|---|
| a) $n = n \% 1000 + n / 100 * 100$ | c) $n = n \% 100 + n / 1000 * 1000$ |
| b) $n = n \% 10 + n / 10 * 10 + n / 100 * 100$ | d) $n = n / 10 \% 100 * 1000 + n / 100$ |

2. Pentru a verifica dacă în tabloul unidimensional (**6, 7, 9, 11, 12, 19, 28**), există elementul cu valoarea **x=18** se aplică metoda căutării binare. Succesiunea de elemente a căror valoare se compară cu x pe parcursul aplicării metodei este:

- | | |
|---------------|---------------|
| a) 11, 9, 6 | c) 11, 19, 28 |
| b) 11, 19, 12 | d) 11, 12, 19 |

3. Se consideră două tablouri unidimensionale **A = (1, 3, 5, 9, 10)** și **B = (2, 4, 6, 7)**. În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:

- | | |
|---------------------------------|--------------------------------------|
| a) (1, 2, 3, 4, 5, 6, 9, 7, 10) | c) Nu se poate realiza interclasarea |
| b) (1, 2, 3, 4, 5, 6, 7, 9, 10) | d) (1, 3, 5, 9, 10, 2, 4, 6, 7) |

4. Care dintre expresiile C/C++ de mai jos are valoarea 1?

- a) $2.17 \geq \text{floor}(2.17) \ \&\& \ 2.17 \leq \text{ceil}(2.17)$
 b) $2.17 \leq \text{floor}(2.17) \ \&\& \ 2.17 + 1 == \text{ceil}(2.17)$
 c) $2.17 < \text{floor}(2.17) \ \&\& \ (2.17) > 1 + \text{floor}(2.17)$
 d) $2.17 > \text{floor}(20.17) \ \&\& \ 2.17 - 1 == \text{floor}(20.17)$

5. Câte atribuiri se execută în total, în secvența alăturată, dacă **n** și **p** sunt variabile de tip întreg?

- | | |
|------|-------|
| a) 7 | c) 9 |
| b) 8 | d) 10 |

```

p=1; n=52;
while(n>=p)
p=p*2;
```

SUBIECTUL al II-lea**(40 de puncte)****1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu $a\%b$ restul împărțirii numărului natural a la numărul natural b .

- a) Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele **4, 3, 12, 36, 45, 51, 27, 87, 17, 25**. **(6p.)**
- b) Dacă pentru n se citește 2, iar pentru p se citește 5, scrieți patru numere distincte care pot fi citite, astfel încât în urma executării algoritmului, valoarea afișată să fie **0**. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască structura **repetă...până când** cu o structură de tip **pentru...execută**. **(6p.)**

```

citește n, p
(numere naturale nenule)
nr ← 0
repetă
    citește x, y
    (numere naturale)
    cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    ■ dacă (x=p) atunci
        nr ← nr + 1
    ■
n ← n - 1
până când n = 0
scrie nr

```

2. Variabilele reale **nota1**, **nota2** și **nota3** memorează notele obținute de un elev la cele trei probe ale examenului de bacalaureat. Declarați corespunzător variabilele și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **reusit** și media obținută, dacă elevul este promovât la examen, sau mesajul **respins** în caz contrar (un elev este promovât dacă are cel puțin nota 5 la fiecare probă și media cel puțin 6).

(6p.)

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg. Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze numerele de mai jos, în această ordine.

(6p.)

for(i=1;i<=4;i++){	0	1	1	1	1	1
for (j=1;j<=6;j++)	1	0	1	0	1	0
.....}	0	1	0	3	3	3
	1	2	3	0	1	2

SUBIECTUL al III-lea**(30 de puncte)**

1. Un număr natural n , având un număr de k cifre, se numește număr **Armstrong** dacă este egal cu suma cifrelor sale ridicate la puterea k .

Se citește un număr natural n , și se cere să se scrie valoarea **1** dacă n este număr Armstrong sau valoarea **0** în caz contrar. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate.

(10p.)

Exemplu: dacă $n=153$ se scrie valoarea 1 ($153=1^3+5^3+3^3$).

2. Scrieți un program C/C++, care citește de la tastatură un număr natural, n ($n \in [2, 10^2]$), apoi cele n elemente ale unui tablou unidimensional, numere naturale. Programul transformă în memorie tabloul, eliminând din componența sa toate numerele care sunt puteri ale lui 2, apoi afișează pe ecran elementele tabloului astfel obținut. Dacă nu există niciun astfel de număr, se va afișa mesajul **nu exista**. Dacă toate elementele tabloului sunt puteri ale lui 2, se va afișa mesajul **s-au eliminat toate elementele**.

(10p.)

Exemplu: pentru $n=10$ și tabloul (23, **32**, 45, 56, **64**, 11, 25, **128**, 78, **8**) după execuția programului tabloul va fi (23, 45, 56, 11, 25, 78).

3. Numerele întregi pozitive cu proprietatea că, prin însumarea iterativă a pătratelor cifrelor lor, se ajunge în cele din urmă la numărul 1, se numesc **numere fericite**. Numărul 7 este un număr fericit pentru că $7^2=49$, $4^2 + 9^2=97$, $9^2 + 7^2=130$, $1^2 + 3^2 + 0^2=10$, $1^2 + 0^2=1$. Prelucrând astfel orice număr, în cele din urmă se va ajunge doar la unul dintre următoarele numere posibile: 0, 1, 4, 16, 20, 37, 42, 58, 89 sau 145.

Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze în fișierul **bac.out**, pe câte un rând, numerele din șir care sunt *fericite* urmate de numărul de iterații necesare pentru a ajunge la numărul 1. Dacă în șir nu există niciun număr fericit se va afișa mesajul **nu exista**. Proiectați un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate.

Exemplu:

bac.in	bac.out
7 13 95 104 86 17 379 226 445 33	7 5
	13 2
	86 2
	379 6
	226 5

- a) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**
- b) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

REZOLVARE SUBIECTE

TEST 5

Subiectul I

1. c
2. b
3. b
4. a
5. b

Subiectul II

1. Rezolvare:

a) 2

n	p	nr	x	y	z
4	3	0	12	36	12
			36	12	0
			12	0	
3		1	45	51	45
			51	45	6
			45	6	3
			6	3	0
			3	0	
2		2	27	87	27
			87	27	6
			27	6	3
			6	3	0
			3	0	
1			17	25	17
			25	17	8
			17	8	1
			8	1	0
			1	0	
0					

b) Algoritmul numără câte seturi de numere x, y au cel mai mare divizor comun numărul p .

Un set de numere posibile: 64, 26, 13, 29

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    int n,p,nr,x,y,z;
    cin>>n>>p;
    nr=0;
    do{
        cin>>x>>y;
        while(y!=0){
            z=x%y;
            x=y;
            y=z;
        }
        if(x==p)
            nr++;
        n--;
    }while(n!=0);
    cout<<nr;
    return 0;
}
```

d) Pseudocod echivalent:

```
citește n, p (numere naturale)
nr ← 0
pentru i ← 1, n, 1 execută
    citește x, y (numere naturale)
    cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    dacă (x = p) atunci
        nr ← nr + 1
scrie nr
```

2. float nota1, nota2, nota3, media;


```
if(nota1>=5 && nota2>=5 && nota3>=5){
    medie=(nota1+nota2+nota3)/3;
    if(medie>=6)
        cout<<"reusit "<<medie;
    else
        cout<<"respins";
}
else cout<<"respins";
```
3.

```
for(i=1;i<=4;i++){
    for(j=1;j<=6;j++){
        if(i%2==1) cout<<i%j<<' ';
        else cout<<j%i<<' ';
    }
    cout<<endl;}
```

Subiectul III

1. Rezolvare:

```

citește n (număr natural)
k ← 0
aux ← n
cât timp (aux ≠ 0) execută
    k ← k + 1
    aux ← [aux / 10]
aux ← n
s ← 0
cât timp (aux ≠ 0) execută
    p ← 1
    pentru i ← 1, k, 1 execută
        p ← p * (aux % 10)
    s ← s + p
    aux ← aux / 10
dacă (n = s) atunci
    scrie 1
altfel
    scrie 0

```

2. Rezolvare:

```

#include <iostream>
using namespace std;
int main(){
    int n, i, j, v[101], aux, ok=0;
    cin >> n;
    for(i=1; i<=n; i++){
        cin >> v[i];
    }
    for(i=1; i<=n; i++){
        aux = v[i];
        while(aux % 2 == 0){
            aux = aux / 2;
        }
        if(aux == 1){
            ok = 1;
            for(j=i; j<=n-1; j++){
                v[j] = v[j+1];
            }
            n--; i--;
        }
    }
    if(ok == 0) cout << "nu exista";
    else{
        if(n == 0)
            cout << "s-au eliminat toate elementele";
        else
            for(i=1; i<=n; i++)
                cout << v[i] << " ";
    }
    return 0;
}

```


3. Rezolvare:

```
#include <iostream>
#include<fstream>
using namespace std;
int main(){
    ofstream fout("bac.out");
    ifstream fin ("bac.in");
    int s,x,it,aux,ok;
    ok=0;
    while(fin>>x){
        aux=x; it=0;
        do{
            s=0; it++;
            while(x!=0){
                s=s+x%10*(x%10);
                x=x/10;
            }
            x=s;
        }while(s!=1 && s!=0 && s!=4 && s!=16 && s!=20 && s!=37 && s!=42
&& s!=58 && s!=89 && s!=145);
        if(s==1){
            fout<<aux<<' '<<it<<endl;
            ok=1;
        }
    }
    if(ok==0)
        fout<<"nu exista";
    return 0;
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 5**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului

SUBIECTUL I**(20 de puncte)**

1c 2b 3b 4a 5b	5 x 4p.
-----------------------	----------------

SUBIECTUL al II-lea**(40 de puncte)**

1.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare set de numere x, y conform cerinței (un set de numere x, y este corect dacă c.m.m.d.c (x,y) este diferit de 5).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -definire a variabilelor conform cerinței -verificare a condiției impuse (*) -afișare mesaj -corectitudine globală a expresiei ¹⁾	6p. 1p. 3p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței referitor la condiția impusă (note, medie, operatori logici utilizați conform cerinței).
3.	Pentru rezolvare corectă -atribuire a valorilor indicate pentru afișare (*) -afișarea valorilor pe linii, conform cerinței -corectitudine globală a secvenței ¹⁾	6p. 4p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (afișare valori, identificare linii cu indice par/linii cu indice impar). O soluție posibilă este afișarea valorii expresiei $i\%j$ elementelor aflate pe linii cu indice impar, respectiv $j\%i$ elementelor aflate pe linii cu indice par.

SUBIECTUL al III-lea**(30 de puncte)**

1.	Pentru algoritm corect -citire a datelor -verificare a proprietății cerute (*) -scriere a datelor -scriere principial corectă a structurilor de control, corectitudine globală a algoritmului ¹⁾ (**) 	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unei cifre, ridicarea unui număr la o putere, calcul sumă). (**) Se va puncta orice formă corectă de structură repetitivă sau decizională.
2.	Pentru program corect -declarare a unei variabile care să memoreze un tablou unidimensional -citire a datelor -transformare a tabloului (*) -afișare a datelor și tratare a cazurilor nu exista/s-au eliminat toate elementele -declarare a variabilelor simple, corectitudine globală a programului ¹⁾ 	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (eliminarea a unui element, algoritm principial corect de verificare a unui număr dacă este putere a lui 2, transformare în memorie).
3.	a) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii/scrierii, citire/scriere în fișier -determinare a valorilor cerute (*), (**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului b) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență	8p. 1p. 5p. 1p. 1p. 2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă parcurge șirul din fișier, memorând valoarea curentă, calculează iterativ suma pătratelor cifrelor cât timp nu s-a ajuns la o valoare particulară și numără iterațiile efectuate.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

MODEL TEST 6

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **x** memorează un număr natural de exact 6 cifre. Care dintre următoarele instrucțiuni C/C++ elimină din **x** cele două cifre situate la mijlocul numărului?

- a. **$x/10000*100+x\%100$** ; b. **$x=x/100-x/100\%100+x\%100$** ;
 c. **$x=x/10000+x\%100$** ; d. **$x\%10000+x/100$** ;

2. Care este rezultatul evaluării expresiei C/C++ următoare: **$14/3*4*3/4$** ?

- a **12.99**
 b **13.99**
 c **12**
 d **14**

3. Ce va afișa pe ecran următorul program?

```
#include <iostream>
using namespace std;
int main()
{int n=10, i;
 int a[]={1,22,33,4,5,6,7,8,9,11};
 for (i=n/2; i; i--) a[i+n/2-1]=a[i];
 for (i=0; i<n; i++) cout<<a[i];
 return 0;
}
```

- a. **1223345678911** b. **12233452233456** c. **12233451223345** d. **22334561223345**

4. Fie secvența:

```
for (i = 0; i < 20; i++) a[i] = i * i - i;
```

Câte din primele 20 de componente ale vectorului **a** memorează valori care au cifra unităților 0?

- a **6**
 b **7**
 c **8**
 d **9**

5. Se consideră secvența de mai jos, în care toate variabilele sunt de tip întreg.

```
i=2;
while(...){if(x%i==0) cout<<i<<" ";
i=i+1;}
```

Pentru a afișa în ordine crescătoare toți divizorii pozitivi ai numărului natural nenul memorat în variabila x , cu excepția lui 1 și a numărului respectiv, o expresie care poate înlocui punctele de suspensie este:

- a $i > 0$
- b $i >= 2$
- c $i <= x/2$
- d $i <= x$

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întreagă a numărului natural x .

a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește numerele **1983** și **237**.

Sc

(6)

p.)

b. Scrieți o pereche de numere care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze valoarea

Sc

11.

(6)

p.)

c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască a doua structură repetitivă cu o structură repetitivă alt de alt tip.

(6p.)

citește a,b (numere naturale)

$x \leftarrow 0$

execută

$x \leftarrow x * 10 + a \% 10 + b \% 10$

$a \leftarrow [a/10]$

$b \leftarrow [b/10]$

cât timp !(a=0 sau b=0)

$n \leftarrow 0$

repetă

$n \leftarrow n * 10 + x / 10 \% 10$

$x \leftarrow [x/100]$

până când

$x = 0$

scrie n

1. În Se consideră două puncte în plan **A** și **B** de coordonate **xa, ya**, respectiv **xb, yb**. Să se scrie o condiție care are valoarea **1** dacă și numai dacă segmentul **AB** este paralel cu axa **OX**. (6p)

2. În secvența de instrucțiuni de mai jos, variabilele **i** și **j** sunt de tip întreg. Scrieți secvența alăturată, înlocuind punctele de suspensie astfel încât, în urma execuției secvenței obținute, să se afișeze pe ecran, în această ordine, numerele de mai jos. (6p.)

2 4 6

2 4

2

for(i=1;i<=4;i++)

{for(.....)

cout<<2*j<<' ';

cout<<endl;

SUBIECTUL al III-lea**(30 de puncte)**

1. Se citesc de la tastatură două numere naturale **n** și **k**. Să se obțină numerele **a** și **b**, unde **a** reprezintă numărul maxim extras din radicalul de ordin **k** al numărului **n**, iar **b** este numărul minim rămas sub radicalul de ordin **k**.

Scrieți în pseudocod algoritmul de rezolvare pentru problema enunțată.

Exemplu: dacă **n=2020** și **k=2**, atunci $\sqrt[2]{2020} = 2\sqrt{505}$, adică **a** va fi **2**, iar **b** va fi **505**.

(10 p.)

2. Scrieți un program Pascal/C/C++ care citește de la tastatură numere naturale din intervalul $[1, 10^9]$, în această ordine **n**, apoi cele **n** elemente ale unui tablou unidimensional. Numărul **n** aparține intervalului $[1, 10^3]$. Programul modifică ordinea valorilor din tablou, astfel încât valorile impare să fie ordonate descrescător, valorile pare să fie ordonate crescător, și fiecare număr de tip par/impar își va păstra indicii originali.

Exemplu: **n=8**, și vectorul: **2, 0, 1, 9, 2, 0, 2, 0**

Vom afișa: **0 0 9 1 0 2 2 2**,

Se observă ca numerele impare si-au păstrat indicii originali, la fel si cele pare.

(10 p.)

3. Fișierul **cifre.in** conține pe prima linie cel mult **10⁶** cifre. Să se citească cifrele din fișier și să se rearanjeze dacă este posibil astfel încât să formeze cel mai mare număr palindrom, sau mesajul **IMPOSIBIL**. Afișarea se va face pe ecran. Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 1 9 2 0 2 0	2 2 1 2 0 2 0
Pe ecran	IMPOSIBIL	2201022

- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2 p.)**
- b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat. **(8 p.)**

REZOLVARE SUBIECTE
TEST 6
SUBIECTUL I
(20 de puncte)

1b	2c	3b	4c	5c	5x4p
----	----	----	----	----	------

SUBIECTUL al II-lea
(40 de puncte)

1. a. După execuția primei structuri repetitive vom obține $x=1121$

După execuția celei de-a doua structuri repetitive vom obține $n=21$

Răspuns 21

a. Orice pereche de numere care să furnizeze după execuția primei structuri repetitive valoarea x de forma $a1b1$ va da răspunsul corect ($0 \leq a, b \leq 9$). O astfel de pereche este: $a=2131$ și $b=5050$

b. Programul C/C++:

```
#include <iostream>
using namespace std;
int main()
{
    int a,b, x, n=0;
    cin>>a>>b;
    x=0;
    do{
        x=x*10+a%10+b%10;
        a=a/10;
        b=b/10;
    }while(a!=0&& b!=0);
    if(x!=0)
    do{
        n=n*10+x/10%10;
        x=x/100;
    }while( x!=0);
    cout<<n;

    return 0;}
```

d. Algoritmul în pseudocod echivalent care înlocuiește a doua structură repetitivă este:
citește a,b (numere naturale)

$x \leftarrow 0$

```
execută
    x ← x*10 + a%10 + b%10
    a ← [a/10]
    b ← [b/10]
```

■ cat timp $a \neq 0$ și $b \neq 0$

$n \leftarrow 0$

```
cat timp  $x \neq 0$  executa
    n ← n*10 + x/10%10
    x ← [x/100]
```

scrie n

2. Condiția care rezolvă cerința: $ya == yb$

3. for (i=1; i<=4; i++)

```
{
    for (j=1; j<=5-i; j++)
        cout<<2*j<<' ';
    cout<<endl;}
```

SUBIECTUL al III - lea

(30 de puncte)

1.

```

Citește n,k
///descompunem in factori primi
a←1
b←1
d←2
cat timp(n>1) executa
|   e←0;
|   cat timp(n%d=0) executa
|   |   e←e+1
|   |   n←[n/d]
|   |   ┌■
|   |   /// modificarea lui a
|   |   Pentru i←1,[e/k] executa
|   |   |   a←a*d
|   |   |   ┌■
|   |   |   /// modificarea lui b
|   |   |   Pentru i←1,e%k executa
|   |   |   |   b←b*d
|   |   |   |   ┌■
|   |   |   |   d←d+1
|   |   └─┘
|   └─┘
└─┘
Scrie a, ' ',b

```

2.

```

#include<fstream>
#include<iostream>
using namespace std;
int n,a[1002],i,j;
int main()
{
    cin>>n;
    for(i=1;i<=n;i++)
        cin>>a[i];
    ///metoda selectiei directe cu testare de paritati
    for(i=1;i<=n;i++)
        for(j=i+1;j<=n;j++)
            if (a[i]%2!=0 and a[j]%2!=0)///ambele impare-sa fie descrescatoare
            {
                if (a[i]<a[j])
                    swap(a[i],a[j]);
            }
            else
            if (a[i]%2==0 and a[j]%2==0)///ambele pare-sa fie crescatoare
            {
                if (a[i]>a[j])
                    swap(a[i],a[j]);
            }
        }
    for(i=1;i<=n;i++)
        cout<<a[i]<<' ';
}

```


3.

a. Fiind doar cifre în fișierul de intrare, vom defini un vector de frecvență de dimensiune 10. Problema nu va avea soluție dacă există 2 sau mai multe cifre cu frecvențe impare. Dacă există soluție, cifrele se vor afișa în două seturi, folosind în fiecare set un număr de cifre egal cu jumătate din frecvența fiecărei cifre. Palindromul de valoare maximă va începe cu un set de cifre alese în sens descrescător, va continua cu cifra cu frecvență impară (dacă există) și se va termina cu al doilea set de cifre alese în sens crescător.

```

#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("cifre.in");
int x,f[10],c,imp,i;
int main()
{
    while(fin>>x)
        f[x]++;
    imp=0;
    for(c=0;c<=9;c++)
        if (f[c]%2==1) imp++;
    if (imp>1) cout<<"IMPOSIBIL";
    else
    {
        //afisam cifrele descrescator jumătate din frecvența..
        for(c=9;c>=0;c--)
            for(i=1;i<=f[c]/2;i++)
                cout<<c;
        //verificam dacă avem o frecv impară
        for(c=0;c<=9;c++)
            if (f[c]%2==1) cout<<c;
        //afisam cifrele crescator jumătate din frecvența..
        for(c=0;c<=9;c++)
            for(i=1;i<=f[c]/2;i++)
                cout<<c;
    }
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 6**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

Răspuns	Punctaj
1b 2c 3b 4c 5c	5x4p.

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 21	6p.	
	b) Răspuns corect: orice pereche de numere care furnizează rezultatul 11.	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are a doua structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat.
2.	Pentru rezolvare corectă ay==by	6p.	
3.	Pentru rezolvare corectă - acces la elementele fiecărei linii - atribuire a valorilor indicate elementelor conform valorilor afișate (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*)Se acordă câte 1p. pentru atribuirea valorilor conform cerinței, pentru elementele situate pe fiecare linie.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL al III - lea

(30 de puncte)

1	Pentru algoritm corect - citirea datelor - determinarea numerelor cerute (*) - scrierea datelor - scriere principial corectă a structurilor de control (**)	10p. 1p. 6p. 1p. 2p	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (factori primi și exponenți + modificare variabila a + modificare variabila b) (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
2	Pentru program corect - declarare a unei variabile care să memoreze un tablou - citire a datelor - memorare a numerelor conform cerinței(*) - afisarea tabloului - declarare a variabilelor simple, corectitudine globală a programului)	10p. 1p. 1p. 6p. 1p. 1p	(*) Se acordă cele 6pct. astfel: - 3p prezentarea metodei de sortare (sel. Directa) - 1p verificare valori impare - 1p verificare valori pare - 1p realizarea interschimbării
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru
	b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului)	8p. 1p. 5p. 1p. 1p.	toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă este folosirea vectorilor de frecvență. Cazul IMPOSIBIL , e îndeplinit dacă avem 2 sau mai multe cifre cu frecvențe impare. Soluția va fi obținută prin afișarea descrescătoare a cifrelor si afișarea crescătoare a cifrelor, in fiecare parte punând jumătate din câte am citit. Caz special ce cifra va fi la mijloc, dacă avem 1 singură frecvență impară?

MODEL TEST 7

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabila **x** memorează un număr natural de exact 5 cifre. Care dintre următoarele expresii are ca valoare un număr natural format din cele 3 cifre situate în mijlocul numărului memorat în variabila **x**?

- a. **$x \% 10$** b. **$x / 10$**
 c. **$x / 10 \% 1000$** d. **$x \% 1000 / 10$**

2. Aplicând algoritmul de căutare binară pentru căutarea unei valori **x** în tabloul unidimensional (2,4,5,6,8,12,16), se fac exact trei comparații. Care dintre următoarele valori poate fi valoarea variabilei **x**?

- a. **2,5** b. **4,12** c. **8,16** d. **2,5,8,16**

3. Ce va afișa pe ecran următorul program?

```
#include <iostream>
using namespace std;
int main()
{   int n=10, i;
    int a[]={11,2,3,44,555,66,7,8,9,1};
    for (i=n/2; i; i--) a[i+n/2-1]=a[i];
    for (i=0; i<n; i++) cout<<a[i];
    return 0;
}
```

- a. **112344555667891** c. **234455566112344555**
 b. **112344555234455566** d. **445552344555661123**

4. Ce va afișa pe ecran următorul program, dacă se citesc de la tastatură valorile {12,1,2,2,3,1,2,1,3,4,1,2,1}?

```
int v[20],n,i,j,k=0;
cout<<"n=";cin>>n;
    for(i=0;i<n;i++)
cin>>v[i];
    for(i=0;i<n-1;i++)
    for(j=i+1;j<n;j++)
    if(v[i]==v[j])
        k++;
cout<<k;
```

- a. **17** b. **20** c. **4** d. **3**

5. Indicați ce valoare are expresia C++ următoare:

$\text{pow}(x,2) - \text{sqrt}(x+9) / \text{abs}(x+3)$, dacă variabila x are valoarea -5 .

- a. 11 b. 11.5 c. 24 d. 0

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.

S-a notat cu $x\%y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[x]$ partea întreagă a numărului natural x .

a. Scrieți valoarea afișată în urma executării algoritmului dacă se citește numărul **29357**. (6p.)

b. Scrieți o valoare de **5** cifre care poate fi citită, astfel încât în urma executării algoritmului, să se afișeze **123**. (6p.)

c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)

d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină alt tip de structuri repetitive. (6p.)

citește n (număr natural)

$x \leftarrow 0$

repetă

$x \leftarrow x * 10 + n \% 10$

$n \leftarrow [n / 10]$

până când $n = 0$

repetă

$n \leftarrow n * 10 + x \% 10$

$x \leftarrow [x / 100]$

până când $x = 0$

scrie n

2. Două puncte **A** și **B** din planul xOy sunt date prin coordonatele lor carteziane X_A, Y_A , respectiv X_B, Y_B . Scrieți o expresie C++ care are valoarea **1** dacă și numai dacă punctele **A** și **B** se află pe prima bisectoare a sistemului de coordonate. (6p.)

3. În secvența de instrucțiuni de mai jos, variabilele i și j sunt de tip întreg. Scrieți secvența alăturată, înlocuind punctele de suspensie astfel încât, în urma execuției secvenței obținute, să se afișeze pe ecran, în această ordine, numerele de mai jos.

2 4 6 8

4 6 8

6 8

8

for($i=1; i \leq 4; i++$)

{

for(.....) cout << ' ';

for(.....)

cout << $2 * j$ << ' ';

cout << endl;

}

SUBIECTUL al III-lea**(30 de puncte)**

1. Se citesc de la tastatură 3 numere:

- **n**, un număr de cel mult 6 cifre scris în baza **b1**
- **b1**, baza numărului original **n**
- **b2**, baza în care va fi trecut numărul **n**

Scrieți un algoritm în pseudocod care să treacă numărul original **n**, din baza **b1** în baza **b2**.

Exemplu: dacă **n=2020**, **b1=3** și **b2=4**, în baza 10, **n** va fi **60**, iar valoarea determinată va fi **330**. **(10p.)**

2. Scrieți un program C/C++ care citește de la tastatură numere naturale din intervalul $[1, 10^9]$, în această ordine: **n**, apoi cele **n** elemente ale unui tablou unidimensional. Numărul **n** aparține intervalului $[1, 10^3]$. Programul modifică ordinea valorilor din tablou, astfel încât valorile impare să fie ordonate crescător, valorile pare să fie ordonate descrescător și fiecare număr de tip par/impar își va păstra indicii originali.

Exemplu: $n=8$, și vectorul: **2, 0, 9, 1, 2, 0, 2, 0**

Vom afișa: **2 2 1 9 2 0 0 0**

Se observă ca numerele impare si-au păstrat indicii originali, la fel si cele pare. **(10p.)**

3. Fișierul **cifre.in** conține pe prima linie cel mult 10^6 cifre. Să se citească cifrele din fișier și să se determine prima secvență cu valori identice de lungime maximă. Secvența va fi identificată prin poziția de început, poziția de final și valoarea cifrei din secvență. Afișarea se va face pe ecran.

Proiectați un algoritm eficient din punct de vedere al spațiului de memorie și al timpului de executare.

Exemple:

Cifre.in	2 0 2 0 0 1 2 2	2 0 1 9 2 0 2 0
Pe ecran	4 5 0	1 1 2

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b. Scrieți programul Pascal/C/C++ corespunzător algoritmului proiectat. **(8p.)**

REZOLVARE SUBIECTE
TEST 7
SUBIECTUL I

(20 de puncte)

1c	2d	3b	4a	5c	5x4p
----	----	----	----	----	------

SUBIECTUL al II-lea

(40 de puncte)

1a. 237

b. orice număr de 5 cifre care are **prima** cifră 1, **a treia** cifră 2 și **a cincea** cifră 3.
(de exemplu 10203)

c.

```
#include <iostream>
using namespace std;
int main()
{ int n,x;
  cin>>n;
  x=0;
  do
  { x=x*10+n%10;
    n=n/10;
  } while (n!=0);
  do
  { n=n*10+x%10;
    x=x/100;
  } while (x!=0);

  cout<<n;
  return 0;
}
```

d.

```
citește n (număr natural)
x←0
┌cât timp n≠0 execută
│ x← x*10+n%10
│ n←[n/10]
└─┘
┌cât timp x≠0 execută
│ n← n*10+n%10
│ x←[x/100]
└─┘
scrie n
```

2.XA==YA && XB==YB

3.

```
for (i=1;i<=4;i++)
{
  for (j=1;j<i;j++) cout<<' ';
  for (j=i;j<=4;j++)
    cout<<2*j<<' ';
  cout<<endl;
}
```

SUBIECTUL al III - lea**(30 de puncte)**

1.

```

Citește n,b1,b2
//formarea numarului din baza b1 in baza 10
p←1;
x←0;
Cat timp (n>0) executa
|   x←x+(n%10)*p //fiecare cifra se inmulteste cu b1^0, b1^1...
|   p←p*b1
|   n←[n/10]
|_■
//numarul x, il impartim la b2
//cu resturile sale formam numar cu fixare de cifre in fata lui
m←0
p←1
cat timp (x>0) executa
|   m←m+(x%b2)*p
|   p←p*10
|   x←[x/b2]
|_■
scrie m;

```

2.

```

#include<fstream>
#include<iostream>
using namespace std;
int n,a[1002],i,j;
int main()
{
  cin>>n;
  for(i=1;i<=n;i++)
    cin>>a[i];
  ///metoda selecției directe cu testare de parităi
  for(i=1;i<=n;i++)
    for(j=i+1;j<=n;j++)
      if (a[i]%2!=0 and a[j]%2!=0)//ambele impare-sa fie crescătoare
      {
        if (a[i]>a[j])
          swap(a[i],a[j]);
      }
      else
      if (a[i]%2==0 and a[j]%2==0)//ambele pare-sa fie descrescătoare
      {
        if (a[i]<a[j])
          swap(a[i],a[j]);
      }
  for(i=1;i<=n;i++)
    cout<<a[i]<<' ';
}

```


3.

a. Se citesc numerele din fișier și în timpul citirii se compară numărul curent y cu cel anterior x . În caz de egalitate se va crește lungimea, iar în caz negativ înseamnă ca am terminat o secvență. Fiecare secvență terminată în timpul citirii dar și la final de citire va fi testată pe baza lungimii ei, comparând-o cu variabila $lmax$, actualizând pe parcurs, poziția de început și final alături de valoarea din secvența maximă. Algoritmul este eficient ca timp de executare întrucât este liniar și este eficient ca spațiu de memorare întrucât nu folosește tablouri.

b.

```

#include<fstream>
#include<iostream>
using namespace std;
ifstream fin("cifre.in");
int x,l,poz,xmax,lmax,st,dr,y;
int main()
{
    fin>>x;
    //prima secventa cu 1 element
    l=1;
    poz=1;
    lmax=0;
    while(fin>>y)
    {
        poz++;
        if (x==y) l++;
        else
        {
            //s-a incheiat o secventa
            if (l>lmax)
            {
                lmax=l;
                xmax=x;
                st=poz-l;//poz minus l(variabila)
                dr=poz-1;//poz minus 1(unu, constanta)
            }
            //pregatim noua secv
            l=1;
            x=y;
        }
    }
    //ultima secv se testeaza
    if (l>lmax)
    {
        lmax=l;
        xmax=x;
        st=poz-l+1;
        dr=poz;
    }
    cout<<st<<' '<<dr<<' '<<xmax;
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 7**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(30 de puncte)**

Răspuns	Punctaj
1c 2c 3b 4a 5d	5x4p.

SUBIECTUL al III-lea**(40 de puncte)**

1.	a) Răspuns corect: 237	6p.	
	b) Răspuns corect: orice număr de 5 cifre care are prima cifră 1, a treia cifră 2 și a cincea cifră 3. (de exemplu 10203)	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principal corectă, dar nu este echivalent cu cel dat. Se acordă numai 4p. dacă se folosesc tot structuri repetitive cu test final.
2.	Pentru rezolvare corectă - acces la câmpurile de pe primul nivel al înregistrării - acces la câmpurile de pe al doilea nivel al înregistrării - afișare conform condiției impuse(*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 1p. 3p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă - acces la elementele fiecărei linii - atribuire a valorilor indicate elementelor conform valorilor afișate (*) - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*)Se acordă câte 1p. pentru atribuirea valorilor conform cerinței, pentru elementele situate pe fiecare linie.

1) Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL al III-lea
(30 de puncte)

1	Pentru algoritm corect - citirea datelor - determinarea numerelor cerute (*) - scrierea datelor - scriere principial corectă a structurilor de control (**) 	10p. 1p. 6p. 1p. 2p	(*) Se acordă câte 3p. pentru: - Transformarea din baza b1 în baza 10 (1pct -obținerea cifrelor, 2 pct-formarea nr) - Transformarea din baza 10 in baza b2 (1pct -obținerea cifrelor, 2 pct-formarea nr) (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
2	Pentru program corect - declarare a unei variabile care să memoreze un tablou - citire a datelor - memorare a numerelor conform cerinței(*) - afisarea tabloului - declarare a variabilelor simple, corectitudine globală a programului1) 	10p. 1p. 1p. 6p. 1p. 1p	(*) Se acordă cele 6pct. astfel: - 3p prezentarea metodei de sortare (sel. Directa) - 1p verificare valori impare - 1p verificare valori pare - 1p realizarea interschimbării
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorii cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului1) 	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. Contorizam lungimea secvenței dacă numărul citit coincide cu anteriorul, la terminarea unei secvențe sau a întregii citiri, comparăm lungimea secvenței cu lungimea maximă. Nu se vor folosi vectori, citire și prelucrare în același timp. Complexitate liniară.

MODEL TEST 8

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Care dintre următoarele expresii are valoarea **0** pentru : **a=5; b=3; c=1; d=3;**
 - a) **(a<b) || c**
 - b) **((b==d) && c) || (a>=b)**
 - c) **c && (d>b)**
 - d) **(a>b) || !(d>a)**

2. Care dintre următoarele atribuiri fac ca valoarea variabilei reale **x** să aibă partea fracționară egală cu **0.0**, indiferent de valoarea inițială a acesteia? (Operația parte întreagă este desemnată prin operatorul [])
 - a) **x ← x * 10;**
 - b) **x ← [x] * 10;**
 - c) **x ← x / 10;**
 - d) **x ← [x] + 10.03;**

3. Determinați valoarea expresiei următoare: **fabs(-11.2) + sqrt(floor(16.23))**
 - a) **-9.2**
 - b) **15**
 - c) **14.2**
 - d) **15.2**

4. Știind că variabilele caracter **x** și **y** au valorile **x='*' și y='-'**, ce se va afișa în urma executării următoarei secvențe de instrucțiuni:


```

a=1; b=12; c=4;
if ( a>b) {
    if (x == '*' ) a*=3; }
else {
    if ( y== '-' ) b -= 3;
    else a += 3;
}
cout<<a<< ' '<<b<< ' '<<c;
      
```

 - a) **13 12 4**
 - b) **3 12 4**
 - c) **1 9 4**
 - d) **3 9 4**

5. Care dintre următoarele numere reprezintă un număr întreg din vocabularul limbajului C/C++?
 a) -2020. b) 2020 c) 23E2 d) '2020'

SUBIECTUL al II-lea
(40 de puncte)
1. Algoritmul alăturat este reprezentat în pseudocod.

- a) Ce valori vor fi afișate dacă se vor introduce, în ordine, valorile: **2, 3, 4, 5, 0**. (6p.)
- b) Determinați un set de date de intrare pentru care ultimele două numere afișate să fie egale. (6p.)
- c) Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d) Modificați algoritmul, fără să introduceți noi variabile sau instrucțiuni, astfel încât să se afișeze succesiv media aritmetică a numerelor citite. (6p.)

```

i ← 0; p ← 1
citește x (x număr întreg)
cât timp x ≠ 0 execută
    i ← i + 1
    p ← p * x
    citește x
    scrie p
Scrie i
  
```

2. Variabilele întregi **x** și **y** memorează câte un număr natural, reprezentând abscisa, respectiv ordonata unui punct, în sistemul de coordonate **xOy**.

Scrieți o secvență de instrucțiuni C/C++ prin care se afișează pe ecran mesajul **axa Ox** dacă punctul se află pe axa **Ox** a sistemului de coordonate, mesajul **axa Oy** dacă punctul se află pe axa **Oy**, mesajul **în originea sistemului xOy**, dacă punctul cu aceste coordonate se află în originea sistemului **xOy**, sau mesajul **în afara axelor**, altfel.

(6p.)

3. Într-un șir de numere întregi primii doi termeni sunt **f₁=1**, respectiv **f₂=1**, iar cel de-al **n**-lea (**n>2**) termen se calculează cu ajutorul formulei **f_n=f_{n-1}+f_{n-2}**.

În secvența alăturată variabilele **a, b, c** sunt de tip întreg.

Scrieți secvența înlocuind zona punctată astfel încât, în urma executării secvenței obținute, variabila întregă **c** să memoreze al **2020**-lea termen al șirului. (6p.)

```

a=1; b=2; c=0;
for(n=3; n<=2020; n++)
{
    ...
}
  
```

SUBIECTUL al III-lea

(30 de puncte)

1. Se citește un număr natural, n , și se cere să se scrie numărul obținut prin înlocuirea fiecărei cifre prime a lui n cu cifra ce reprezintă dublul cifrei respective. Dacă nu este posibil acest lucru, cifra primă va fi eliminată din număr. Dacă nu există nici o cifră primă în scrierea numărului, se va afișa valoarea 0 . Scrieți, în pseudocod, algoritmul de rezolvare pentru problema enunțată.

Exemplu: dacă $n=35670$, atunci se scrie numărul 660 , iar dacă se citește numărul 46 , se afișează valoarea 0 . (10p.)

2. Scrieți un program C/C++ care citește de la tastatură un număr natural par, n ($4 \leq n \leq 22$), apoi cele n elemente ale unui tablou unidimensional, numere reale. Programul afișează pe ecran toate perechile de numere egal depărtate de extremitățile tabloului, care au aceeași parte întreagă. Perechile vor fi încadrate între paranteze, numerele fiind separate prin virgulă.

Exemplu: pentru $n=6$ și tabloul $(2.32, 4.34, 9.2, 1.0, 4.05, 2.45)$ se afișează perechile: $(2.32, 2.45)$ $(4.34, 4.05)$ (10p.)

3. Fișierul **bac.txt** conține un șir format din cel puțin două și cel mult 10^6 numere naturale din intervalul $[100, 999]$, separate prin câte un spațiu. Se cere să se afișeze în fișierul **bac.out**, pe linii diferite, în ordine crescătoare, numerele citite, grupate după cifra dominantă (prima din scrierea zecimală).

Numerele de pe aceeași linie se afișează separate prin câte un spațiu.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.

Exemplu: dacă fișierul conține numerele **334 122 211 347 122 156 395** conținutul fișierului **bac.out** va fi următorul

122 122 156

211

334 347 395

- a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. (2p.)
b) Scrieți programul C/C++ corespunzător algoritmului descris. (8p.)

REZOLVARE SUBIECTE
TEST 8
SUBIECTUL I

(20 de puncte)

1. c
2. b
3. d
4. c
5. b

SUBIECTULal- II -lea

(40 de puncte)

1. a) 2 6 24 120 4

Rezolvare:

i	p	x
0	1	2
1	2	3
2	6	4
3	24	5
4	120	0

- b) 1 2 0 - o soluție posibilă

- c) Programul C++ :

```
#include <iostream>
using namespace std;
int main()
{int x, p, i;
  i=0; p=1;
  cin>>x;
  while(x)
  {i=i+1;
  p=p*x;
  cin>>x;
  cout<<p;}
  cout<<i;
  return 0;
}
```

- d) Pseudocod echivalent:

```
i ← 0; p ← 1
citește x (x număr întreg)
cât timp x ≠ 0 execută
  i ← i + 1
  p ← p * x
  citește x
  scrie p/i
scrie i
```

2. Rezolvare:

```
if(x==0&& y==0) cout<<"in originea sistemului xOy";
else
if(y==0) cout<<"axa Ox";
else if(x==0) cout<<"axa Oy";
else cout<<"in afara axelor";
```

3. Rezolvare:

```

a=1; b=2;
for(n=3;n<=2020;n++)
{c=a+b;
a=b;
b=c;
}

```

SUBIECTUL al III-lea

(30 de puncte)

1. Rezolvare:

```

citeste n; cn←n
nr←0; p←1
cât_timp n≠0 execută
  c←n%10
  dacă (c=2 sau c=3) atunci nr←nr+2*c*p; p←p*10
    altfel dacă (c≠5 și c≠7) atunci
      nr←nr+c*p; p←p*10
    sfârșit_dacă
  sfârșit_dacă
  n←n/10
sfârșit_cât_timp
dacă nr=cn atunci nr←0
sfârșit_dacă
scrie nr

```

2. Rezolvare:

```

#include <iostream>
using namespace std;
int main()
{int n,i; float v[23];
cin>>n;
for(i=1;i<=n;i++)cin>>v[i];
for(i=1;i<=n/2;i++)if((int)v[i]==(int)v[n-i+1])
  cout<<'('<<<v[i]<<<','<<<v[n-i+1]<<<')';
return 0;
}

```

3. O soluție posibilă:

```

#include <fstream>
using namespace std;
int v[1000];
int main()
{ ifstream f("bac.txt");
  ofstream g("bac.out");
  int i,j,x,k,ok;
  while(f>>x) v[x]++;
  for(j=1;j<=9;j++)
  {ok=1;
  for(i=j*100;i<=j*100+99;i++)

    for(k=1;k<=v[i];k++)
      { g<<i<<' ';ok=0;}
  if (ok==0)
    g<<endl;
  }
return 0;
}

```


BAREM DE EVALUARE ȘI DE NOTARE**TEST 8**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

1. c	2. b	3. d	4. c	5. b	5x4p.
-------------	-------------	-------------	-------------	-------------	--------------

SUBIECTUL al II - lea**(40 de puncte)**

1.	a) Răspuns corect: 2 6 24 120 4	6p.	
	b) Pentru răspuns corect	6p.	
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiune repetitivă - atribuirii - corectitudine globală a programului ¹⁾	10p. 1p. 2p. 2p.(*) 2p. 2p. 1p.	(*) Se acordă numai 1p dacă s-au afișat doar valorile din interiorul structurii repetitive
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	
2.	- Pentru rezolvare corectă* - corectitudine globală a secvenței	4p. 2p.	(*) Se acordă câte 1p. pentru fiecare aspect: (poziționare în originea sistemului, axa Ox, axa Oy, în afara axelor)
3.	Pentru rezolvare corectă	6p.	

1.	Pentru algoritm corect - citirea numărului - inițializări variabile - determinare a numărului cerut (**) - tratarea cazului particular - corectitudine globală a algoritmului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p.	(**) Se acordă câte 2p. pentru fiecare aspect al cerinței (testarea cifrelor prime ce pot fi dublate, cifrele prime ce vor fi eliminate, corectitudinea numărului construit).
2.	Pentru program corect - declarare a unei variabile care să memoreze un șir de numere reale - citire a datelor - determinare perechi cu proprietatea ceruta (*) - afișare a datelor - declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p 1p 1p 6p 1p 1p	(*) Se acordă câte 2p pentru fiecare aspect al cerinței (identificare numere egal departate de centru, verificarea proprietatii cerute, parcurgere corecta a sirului)
	a) Pentru răspuns corect - coerența descrierii algoritmului (*) - justificare a unor elemente de eficiență	2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient.
3.	b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier, afișare în fișier - afișarea corectă a valorilor citite, grupate după cifra dominantă (*),(**) - utilizarea unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	8p. 2p. 4p. 1p. 1p.	(**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul corect pentru cazul în care valorile citite se repetă. (***) Se acordă punctajul numai pentru un algoritm liniar . Un astfel de algoritm folosește vectorul de frecvență, elementele citite având exact trei cifre.

1) Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

MODEL TEST 9

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele x , y și z sunt de tip întreg și memorează numere naturale din intervalul $[1, 10^3]$. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul memorat în variabila întregă x NU aparține reuniunii de intervale $[-5, -1] \cup [1, 5]$.

a. **!(x>=-5 && x<=-1) || !(x>=1 && x<=5)**

b. **!(x>=-5 || x<=-1 || x>=1 || x<=5)**

c. **x < -5 || x > 5 || x > -1 && x < 1**

d. **x < -5 && x > 5 && x > -1 || x < 1**

2. Pentru secvența de program alăturată, se dau mai jos patru triplete de numere. Fiecare dintre aceste triplete reprezintă un set de valori pentru variabila de intrare n . Care dintre triplete are proprietatea că, pentru toate cele trei valori ale lui n din triplet se obține aceeași valoare a lui S ?

```
S=0; i=3;
while (i<=n)
{ S=S+i;
  i=i+3;
}
cout<<S; | printf("%d",S);
```

a. **(3, 5, 6)**

b. **(6, 7, 8)**

c. **(10, 11, 12)**

d. **(6, 9, 12)**

2. Variabilele x și y sunt de tip real. O transcriere în limbajul C/C++ a expresiei alăturate este:

$$\sqrt{x^2 + 1}$$

a. **sqrt((1+x)*x)**

b. **1+sqrt(x)**

c. **sqrt(1+x*x)**

d. **(1+x*x)^{1/2}**

3. Deduceți care vor fi elementele vectorului a la sfârșitul execuției secvenței de program alăturate, pentru cazul în care $n=7$ și elementele vectorului v sunt, în ordine **(0, 2, 7, 3, 4, 8, 5)**.

```
j=0;
for(i=0;i<n;i++)
if(v[i]%2==0 && i%2!=0)
{ a[j]=v[i]; j=j+1;}
```

a. **(3)**

b. **(2, 8)**

c. **(0, 4)**

d. **(0,7, 4, 5)**

4. Vectorul v are 7 elemente întregi $(-1, 2, -3, 4, 5, -6, 7)$, i și a sunt variabile întregi. După executarea secvenței de program alăturate vectorul v este:

```

a=0;
for (i=0; i<=6; i++)
  if (a>v[i]){a=v[i]; v[i]=0;}
  else v[i]++;

```

- a. 0 3 -2 0 0 -5 0 b. 0 3 0 5 6 0 8 c. 0 0 -2 0 0 -5 d. 0 3 -2 0 0 -5 8

SUBIECTUL al II-lea
(40 puncte)

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $a\%b$ restul împărțirii numărului natural a la numărul natural nenul b .

- a. Ce se va afișa dacă se citește pentru n valoarea 5 și pentru x valorile: 16, 80, 48, 20, 240 (6p.)
- b. Dacă $n=4$, dați exemplu de patru valori pentru x pentru care algoritmul să afișeze 2020. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți un algoritm în pseudocod echivalent cu algoritmul dat în care să se utilizeze doar structuri repetitive condiționate posterior (cu test final). (6p.)

```

citește n (număr natural)
d←0
pentru i←1,n execută
  citește x (număr natural nenul)
  dacă d=0 atunci
    d←x
  altfel
    repetă
      r←x % d
      x←d
      d←r
    pînă când r=0
  d←x
scrie d

```

2. Variabila a memorează un număr natural. Scrieți o secvență de instrucțiuni C++ în urma executării căreia să se afișeze pe ecran suma cifrelor impare din numărul respectiv. (6p.)

3. Fără a utiliza alte variabile, înlocuiți punctele de suspensie din secvența de mai jos astfel încât, în urma executării ei, să se insereze pe poziția p numărul întreg, v , într-un șir s cu n numere întregi, cu elemente numerotate de la 1. (de exemplu, dacă șirul s are elementele 2,4,6,8,10, valoarea $v=12$ și dacă $p=3$ șirul s va avea după execuția secvenței elementele: 2, 4, 12, 6, 8, 10.

```

cin>>p;
cin>>v;
n=n+1;
.....
s[i]=s[i-1]
.....

```

(6p.)

SUBIECTUL al III-lea

(30 de puncte)

1. Scrieți un program care citește de la tastatură un număr natural n , care reprezintă numărul de elemente ale unui tablou unidimensional a cu maximum 100 de numere întregi și apoi cele n elemente ale tabloului. Programul rearanjează elementele tabloului unidimensional astfel încât toate valorile de 2 cifre să fie ordonate descrescător, celelalte elemente din vector nefiind afectate de modificări. Tabloul modificat va fi afișat pe ecran, elementele fiind separate printr-un singur spațiu.

Exemplu: dacă tabloul are 6 elemente și este de forma (12, -7, 61, -32, 800, 7), după apel, acesta va fi: (61, -7, 12, -32, 800, 7).

(10p.)

2. Scrieți în pseudocod algoritmul de rezolvare al problemei următoare: se citește de la tastatură un număr natural n și afișează suma divizorilor primi din descompunerea numărului în factori primi.

Exemplu: dacă $n=1540$ se va afișa $S=2+5+7+11=25$. ($1540=2^2 \cdot 5 \cdot 7 \cdot 11$)

(10p.)

3. Se consideră șirul 1, 1, 2, 1, 2, 3, 1, 2, 3, 4... .. construit astfel: prima grupă este formată din numărul 1, a doua grupă este formată din numerele 1 și 2,.... grupa a k -a, este formată din numerele 1, 2, $k-1$, k .

Se cere să se citească din fișierul **bac.in** un număr natural n ($n \leq 1000$) și să se afișeze în fișierul **bac.out** cel de al n -lea termen al șirului dat. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă valoarea lui n este 10 se va afișa 4; dacă valoarea lui n este 12 se va afișa 2.

a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia.

(2p.)

b) Scrieți programul C/C++ corespunzător algoritmului descris.

(8p.)

REZOLVARE SUBIECTE
TEST 9
SUBIECTUL I

(20 de puncte)

1 c	2 b	3 c	4 b	5 b	5x4p
-----	-----	-----	-----	-----	------

SUBIECTUL al II – lea

(40 de puncte)

1.a. 4

b. oricare 4 numere al căror cmmdc este 2020. Un răspuns corect ar putea fi: 2020, 6060, 10100, 14140

```

c. #include <iostream>
using namespace std;
int main()
{ int n,d,x,r;
  cin>>n;
  d=0;
  for(int i=1;i<=n;i++)
  { cin>>x;
    if(d==0)
      d=x;
    else
    {
      do{
        r=x%d;
        x=d;
        d=r;
      } while(r);
      d=x;
    }
  }
  cout<<d;
  return 0;
}

```

```

d. citește n (număr natural)
d←0
i←1
dacă i<=n atunci
|   repetă
|   |   citește x (număr natural nenul)
|   |   dacă d=0 atunci
|   |   |   d←x
|   |   |   altfel
|   |   |   repetă
|   |   |   |   r←x % d
|   |   |   |   x←d
|   |   |   |   d←r
|   |   |   |   până când r=0
|   |   |   d←x
|   |   └─┬─┘
|   |       └─┬─┘
|   |           i←i+1
|   |           până când i>n
|   └─┬─┘
|       └─┬─┘
|           scrie d

```

2.

```

b=0;
cin >> a;
while (a != 0)
{
  if(a%10%2==1)
    b = b + a % 10;
  a = a / 10;
}
cout << b;

```

3.

```
cin>>p;
cin>>v;
n=n+1;
for (i=n;i>=p+1; i--)
    s[i]=s[i-1];
s[p]=v;
```

SUBIECTUL al III - lea**(30 de puncte)**

```
1. #include <iostream>
    using namespace std;
int main()
{
int a[101], int n, aux;
cin>>n;
for(int i=1;i<=n;i++)
    cin>>a[i];
for(int i=1;i<n;i++)
    if(a[i]/100==0&&a[i]/10!=0)
        for(int j=i+1;j<=n;j++)
            if(a[j]/100==0&&a[j]/10!=0)
                if(a[i]<a[j])
                    { aux=a[i];
                      a[i]=a[j];
                      a[j]=aux;}
for(int i=1;i<=n;i++)
    cout<<a[i]<<" ";
return 0;}
```

2.

```
citeste n;
d←2; S←0;
cat timp (n>1) executa
| nr←0;
| cat timp (n%d=0) executa
| | n←n/d;
| | nr←nr+1;
| | ■
| daca (nr>0) atunci
| | S=S+d;
| | ■
| d←d+1;
| ■
scrie S;
```

3.

Se împarte șirul 1, 1, 2, 1, 2, 3, 1, 2, 3, 4, în grupe:

grupa 1: 1;

grupa 2: 1, 2;

grupa 3: 1, 2, 3;

.....

grupa k: 1, 2, ..., k;

- Presupunem că termenul de rang n este ultimul din grupa completă k; în acest caz se obține relația $1+2+3+\dots+k=n$, adică $k*(k+1)/2=n$
- Numărul de grupe complete ale șirului până la termenul de rang n se obține ca soluție a ecuației $k^2+k-2*n=0$
- Se verifică dacă termenul de rang n este ultimul dintr-o grupă completă sau face parte dintr-o grupa incompletă
- Se determină poziția termenului de rang n în cadrul grupei lui și se afișează, aceasta fiind și valoarea termenului cerut

```
#include <iostream>
```

```
#include<cmath>
```

```
using namespace std;
```

```
int main()
```

```
{ long long n,d,k,p;
```

```
cin>>n;
```

```
d=1+8*n;// se determină discriminantul ecuației de gradul II corespunzătoare șirului
```

```
k=(-1+sqrt(d))/2; // se determină numărul de grupe complete din șir până la termenul de rang n
```

```
if(n==k*(k+1)/2) //dacă termenul de rang n este ultimul din grupa completă k se afișează
```

```
cout<<k;
```

```
else
```

```
{p=n-k*(k+1)/2; //se stabilește poziția termenului de rang n în cadrul grupei lui
```

```
cout<<p; //valoarea termenului este egală cu poziția lui în cadrul grupei din care face parte
```

```
}
```

```
return 0;
```

```
}
```


BAREM DE EVALUARE ȘI DE NOTARE**TEST 9**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 puncte)**

1 c 2 b 3 c 4 b 5 b	5x4p	
----------------------------	-------------	--

SUBIECTUL al II – lea**(40 de puncte)**

1.	a) Răspuns corect: 4	6p	
	b) Răspuns corect: oricare 4 numere al căror cmmdc este 2020	6p	
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiuni de decizie -instrucțiuni repetitive -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 1p. 4p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este corectă.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) -corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă - algoritm corect de spargere a numărului în cifre - identificare cifră impară -determinare a valorii cerute - afișare a rezultatului -corectitudine sevență	6p. 2p. 1p. 1 p 1p 1p	
3.	Pentru rezolvare corectă -acces la un element al tabloului -atribuire a valorilor indicate elementelor tabloului (*) -corectitudine globală a secvenței	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect specific (atribuire valori 1, atribuire valori 2, atribuire valori 3, elemente suport) conform cerinței.

SUBIECTUL al III - lea

(30 puncte)

1.	Pentru program corect -declarare variabile -citire vector -aranjare a elementelor în ordinea cerută (*) -afisare elemente -corectitudine program	10p. 1p. 1p. 6p. 1p. 1p.	(*)Se acordă câte 2p. pentru fiecare aspect specific (identificare a elementelor cu exact 2 cifre, ordonare descrescătoare, pastrarea celorlalte elemente nemodificate)
2.	Pentru program corect -citire număr -determinare a valorii cerute (*) -afișare a rezultatului - corectitudine globală algoritm	10p. 2p. 6p 1p. . 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unui divizor, identificare a unui divizor prim, calculare a sumei)
3.	a) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență b) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier -determinare a valorii cerute (*),(**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului1)	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm care nu folosește nicio structură repetitivă și care folosește doar variabile simple. O soluție posibilă citește valoarea lui n din fișier, stabilește numărul de grupe complete ale șirului, grupa din care face parte termenul de rang n, poziția termenului de rang n în cadrul grupei lui și valoarea termenului Un algoritm posibil de rezolvare este: Citeste n $d \leftarrow 1 + 8 * n$ $k \leftarrow (-1 + \text{sqrt}(d)) / 2$ daca $n = k * (k + 1) / 2$ atunci scrie k altfel $p \leftarrow n - k * (k + 1) / 2$ scrie p ■

MODEL TEST 10

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (**bold**), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.

SUBIECTUL I

(20 puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

1. Variabilele x , y și z sunt de tip întreg și memorează numere naturale din intervalul $[1,1001]$. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă produsul valorilor variabilelor x , y , z este strict mai mare decât zero.

a. $x > 0 \ \&\& \ y > 0 \ \&\& \ z > 0$	b. $!(x * y * z <= 0)$
c. $x * y + y * z + x * z > 0$	d. $x * y > 0 \ \&\& \ y * z > 0 \ \&\& \ x * z > 0$

2. În urma interclasării în ordine crescătoare a tablourilor unidimensionale $A=(1,2,4)$ și $B=(2,3,6,7)$ se obține tabloul:

a. (1,2,2,3,4,6,7)	b. (1,2,3,4,6,7)
c. (1,2,2,3,4,7,6)	d. (7,6,4,3,2,2,1)

3. Variabilele x și y sunt de tip real. O transcriere în limbajul C/C++ a expresiei $\sqrt{x^2(x-y)}$ este:

a. <code>pow(x*x(sqrt(x-y)))</code>	b. <code>sqrt(x*x*x-x*y)</code>
c. <code>pow(x*x*x-x*x*y)</code>	d. <code>sqrt(pow(x,2)*(x-y))</code>

4. Variabilele i și j sunt de tip întreg. Indicați expresia care poate înlocui zona punctată astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran valorile alăturate.

<pre>for(i=0;i<5;i++) { for(j=0;j<5;j++) cout<<.....<<" ";printf("%d ",.....); cout<<endl printf("\n");}</pre>	<pre>25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1</pre>
---	--

a. $5*(4-i) + (5-j)$	b. $5*(4-j)-5*i$	c. $5*(i+1)+1+4*j$	d. $4*(5-i)+5-j$
-----------------------------	-------------------------	---------------------------	-------------------------

5. Fie dat tabloul unidimensional: (2, 3, 5, 7, 8, 9). În urma căutării binare a valorii $x=8$ în tablou, aceasta va fi găsită după un număr de pași egal cu:

a. 1	b. 2	c. 3	d. 4
-------------	-------------	-------------	-------------

SUBIECTUL al II-lea**(40 puncte)**

1. Algoritmul alăturat este reprezentat în pseudocod. S-a notat cu $a \% b$ restul împărțirii numărului natural a la numărul natural nenul b și cu $[c]$ partea întreagă a numărului real c .

- a) Scrieți ce valoare va fi afișată în urma execuției algoritmului dacă se citesc, în această ordine, valorile 11,57,9 . **(6p)**
- b) Dacă pentru c se citește valoarea 17, dați un exemplu de valori pentru a și b astfel încât să se afișeze două valori nule 0 0 . **(6 p)**
- c) Să se scrie programul C/C++ corespunzător algoritmului dat. **(10 p)**
- d) Să se scrie un algoritm echivalent cu cel dat în care să nu se folosească structuri repetitive . **(6 p)**

Citește a, b, c (numere naturale nenule)

```

┌dacă a > b atunci
| a ← a + b
| b ← a - b
| a ← a - b
└─
s ← 0
n ← 0
┌pentru x ← b, a, -1 execută
| ┌dacă x % c = 0 atunci
| | n ← n + 1
| | s ← s + x
| └─
└─
Scrie n, ' ', s

```

2. Scrieti declaratii pentru următoarele variabile de tip intreg : $x_1, y_1, r_1, x_2, y_2, r_2$, reprezentand coordonatele centrelor si razele a doua cercuri c_1 si c_2 . Scrieți o expresie C++ care să fie adevărată dacă și numai dacă cele două cercuri c_1 si c_2 , de centre diferite, nu se intersectează. **(6p.)**

3. Elementele unui tablou unidimensional sunt, în această ordine, (8,6,7,12,11,16,10). Elementele se rearanjează în ordine crescătoare , folosind metoda de ordonare bubble-sort. Precizati numarul de comparatii făcute pentru ordonare.

(6p.)**SUBIECTUL al III-lea****(30 puncte)**

1. Se citesc două numere naturale , a și b ($a \in [0,10^9]$, $b \in [0,10^9]$). Scrieti un program pseudocod care determină si afișează cel mai mare număr care se poate forma cu cifrele distincte comune celor două valori, sau valoarea -1 în cazul în care cele două valori nu au nicio cifră comună. **(10p.)**

Exemplu: Dacă se citesc valorile $a=806528$ și $b=207068$, se afișează numărul 8620 .

2. Se citește de la tastatură un numar natural n ($n \in [0,10^2]$) și apoi n numere naturale reprezentând elementele unui tablou. Să se afișeze pe ecran toate elementele din vector diferite de ultimul , care conțin același număr de cifre pare ca și ultimul element , sau mesajul “nu există” dacă niciun element din tablou nu îndeplinește aceasta condiție. Valorile vor fi scrise pe o linie de ecran, separate cu spațiu . **(10p.)**

Exemplu: Daca se citesc $n=7$ si tabloul (2341, 455, 901, 104, 2228, 6091, 8899), se vor afișa pe ecran valorile 2341 104 6091.

3. Fișierul “**bac.txt**” conține un șir de cel mult 10^9 numere naturale din intervalul $[0,10^9]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran în ordine **crescătoare**, separate cu spațiu, toate valorile distincte de două cifre care apar de cele mai multe ori ca prefixe ale numerelor cu mai mult de două cifre aflate în fișier . Dacă nu există astfel de valori, se va afișa pe ecran mesajul “**nici o valoare**”.

Proiectați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul bac.txt conține valorile 1234 780 1204 978 132 128 7800 222 97 7831
13, se afișează pe ecran valorile 12 78.

a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**

b. Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**

REZOLVARE SUBIECTE**TEST 10****SUBIECTUL I****(20 puncte)**

1. Variabilele x , y și z sunt de tip întreg și memorează numere naturale din intervalul $[1,1001]$. Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă produsul valorilor variabilelor x , y , z este strict mai mare decât zero.

a. $x > 0 \ \&\& \ y > 0 \ \&\& \ z > 0$

b. $!(x * y * z <= 0)$

c. $x * y + y * z + x * z > 0$

d. $x * y > 0 \ \&\& \ y * z > 0 \ \&\& \ x * z > 0$

Rez: Punctul a nu este corect, deoarece există triplete de numere care nu satisfac a, dar pot fi soluții (ex: 1, -2, -3). Punctul c, nu: ex: $x = -1, y = -2, z = -3, xy + xz + yz = 11 > 0$, dar $xyz = -6 < 0$. Punctul d, asemănător cu c. Folosind echivalența: $!(x <= y) \Leftrightarrow x > y$, obținem că expresia de la punctul b. este echivalentă cu: $x * y * z > 0$. Răspuns corect: **b**.

2. În urma interclasării în ordine crescătoare a tablourilor unidimensionale $A = (1, 2, 4)$ și $B = (2, 3, 6, 7)$ se obține tabloul:

a. (1,2,2,3,4,6,7)

b. (1,2,3,4,6,7)

c. (1,2,2,3,4,7,6)

d. (7,6,4,3,2,2,1)

Rez: Tablourile de la pct. a, c. nu sunt sortate crescător. Tabloul de la punctul b nu-l conține pe 2 de două ori, deși acesta apare atât în A, cât și în B (câte o singură apariție). Răspuns corect: **c**.

3. Variabilele x și y sunt de tip real. O transcriere în limbajul C/C++ a expresiei alăturate este:

$$\sqrt{x^2(x-y)}$$

a. $\text{pow}(x * x, \text{sqrt}(x - y))$

b. $\text{sqrt}(x * x * x - x * y)$

c. $\text{pow}(x * x * x - x * x * y)$

d. $\text{sqrt}(\text{pow}(x, 2) * (x - y))$

Rez: Funcția putere - $\text{pow}(a, b) = a^b$ are două argumente, deci variantele a, b, c nu pot fi corecte. Funcția radical - $\text{sqrt}(x) = \sqrt{x}$, are un singur argument. În plus, expresia de la punctul a conține o eroare sintactică: lipsește un operator între al doilea x și $(\text{sqrt}(x - y))$; și $x(\text{sqrt}(x - y))$ nu poate fi nici nume de variabilă sau de constantă. De observat că la punctul d, apelul funcției pow este conținut de argumentul funcției sqrt . Răspuns corect: **d**.

4. Răspuns corect: **a**.

5. Fie dat tabloul unidimensional: (2, 3, 5, 7, 8, 9). În urma căutării binare a valorii $x = 8$ în tablou, aceasta va fi găsită după un număr de pași egal cu:

a. 1

b. 2

c. 3

d. 4

Rez: Dacă notăm cu p și u indicii primului și ultimului element, vom avea inițial $p = 1$ și $u = 6$. La primul pas: calculăm poziția de mijloc: $m = (p + u) / 2 = (1 + 6) / 2 = 3$ și comparăm elementul de pe poziția 3 din tablou, adică pe 5 cu elementul căutat: 8. Cum $8 > 5$, vom căuta pe 8 la pasul 2 în jumătatea dreaptă a tabloului, adică între $p = m + 1 = 3 + 1 = 4$ și $u = 6$ (care rămâne neschimbat). Se calculează din nou $m = (p + u) / 2 = (4 + 6) / 2 = 5$ și se observă că pe poziția $m = 5$ se află chiar elementul căutat: 8.

Răspuns corect: **b**.

SUBIECTUL al II-lea

(40 puncte)

1.a) 5 180

b) Orice doua valori ce definesc extremitatile unui interval ce nu contine niciun multiplu de 17

c)

```

#include <iostream>
using namespace std;
int a,b,c,x,n,s;
int main()
{
    cin>>a>>b>>c;
    if(a>b)
    {
        a=a+b;
        b=a-b;
        a=a-b;
    }
    for(x=b;x>=a;x--)
        if(x%c==0)
        {
            n++;
            s=s+x;
        }
    cout<<n<<' '<<s;
    return 0;
}

```

d)

citește a, b, c (numere naturale nenule)

┌dacă $a > b$ atunci| $a \leftarrow a + b$ | $b \leftarrow a - b$ | $a \leftarrow a - b$

└■

 $n \leftarrow b/c - (a-1)/c$ ┌dacă $a \% c \neq 0$ atunci| $a \leftarrow (a/c + 1) * c$

└■

 $b \leftarrow b/c * c$ $s \leftarrow (a + b) * n / 2$

Scrie n, ' ', s

2. $\sqrt{\text{pow}(x_1-x_2,2)+\text{pow}(y_1-y_2,2)} > r_1+r_2$

3. Se efectueaza 6 interschimbari.

SUBIECTUL al III-lea

(30 puncte)

1. întreg n , c , x , a , b , ok , ca , cb , ka , kb citește a, b $x \leftarrow 0$ $ok \leftarrow 0$ pentru $c \leftarrow 9, 0, -1$ executa $ca \leftarrow a$ $cb \leftarrow b$ $ka \leftarrow 0$ $kb \leftarrow 0$ cât timp $ca \neq 0$ executădacă $ca \% 10 = c$ atunci $ka \leftarrow ka + 1$ $ca \leftarrow ca / 10$ cât timp $cb \neq 0$ executădacă $cb \% 10 = c$ atunci $kb \leftarrow kb + 1$ $cb \leftarrow cb / 10$ daca $ka \neq 0$ si $kb \neq 0$ atunci $x \leftarrow x * 10 + c$ $ok \leftarrow 1$ dacă $ok = 1$ atunciscrie x

altfel

scrie -1

2.

O solutie posibila:

```

#include <iostream>
using namespace std;
int n, v[101], i, aux, ku, ki, ok;
int main()
{
    cin >> n;
    for(i=1; i<=n; i++)
        cin >> v[i];
    ku=0; aux=v[n];
    while(aux)

```



```

{if(aux%2==0)
ku++;
aux=aux/10;
}
for(i=1; i<=n-1; i++)
{ aux=v[i];
ki=0;
while(aux)
{if(aux%2==0)
ki++;
aux=aux/10;
}
if(ku==ki)
{cout<<v[i]<<' ';
ok=1;
}
}
if(ok==0)
cout<<"nu exista ";
return 0;
}

```

3. a) Algoritmul citește secvențial valorile din fișier, contorizează frecvența prefixelor de două cifre în vectorul `fr` și apoi determină maximumul de apariții. Se parcurge vectorul de frecvență și se afișează valorile care au frecvența maximă, în ordine crescătoare.

Algoritmul este eficient ca timp de execuție deoarece face determinările cerute odată cu citirea datelor, deci este liniar.

b) `#include <bits/stdc++.h>`

```

using namespace std;
ifstream fin("bac.txt");

```

```

int fr[100], x, maxi, ok;

```

```

int main()

```

```

{
while(fin>>x)

```

```

    if(x>99)
    {
while(x>99)
x=x/10;
fr[x]++;
}

```

```

for(x=10;x<100;x++)

```

```

    if(fr[x]>maxi)

```

```

maxi=fr[x];
for(x=10;x<100;x++)
if(fr[x]==maxi)

```

```

{
cout<<x<<' ';
ok=1;
}

```

```

}
if(!ok)
cout<<"nici o valoare";
return 0;
}

```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 10**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului.

SUBIECTUL I**(20 de puncte)**

Răspuns	Punctaj
1b 2c 3d 4a 5b	5x4p.

SUBIECTUL III**(40 de puncte)**

1.	a) Răspuns corect: 5 180	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței.
	b) Răspuns corect:	6p.	Se acordă câte 3p. pentru fiecare dintre cele două numere conform cerinței (orice două valori care nu conțin, în intervalul delimitat, niciun multiplu de 17).
	c) Pentru program corect - declarare variabile - citire date - afișare date - instrucțiuni repetitive (*) - atribuiri - corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 5p. 1p. 1p.	(*) Se acordă numai 3p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect - echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p.	(*) Se acordă 2p pentru determinarea numărului de multipli de c din intervalul [a,b] și 4p pentru calculul sumei progresiei aritmetice de rație c, termen inițial a (adus la valoarea primului multiplu de c din interval) și termen final b (adus la valoarea ultimului multiplu de c din interval)
2.	Pentru rezolvare corectă - declarare a variabilelor conform cerinței - afișarea expresiei conform cerinței - corectitudine globală a secvenței ¹⁾	6p. 1p. 4p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect (condiție corectă, afișare pentru fiecare caz) conform cerinței.
3.	Pentru rezolvare corectă	6p.	

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.

SUBIECTUL III**(30 de puncte)**

1	Pentru algoritm corect - citirea datelor - determinare a numărului cerut(*) - scrierea datelor - scriere principial corectă a structurilor de control (**) 	10p 1p 6p. 1p. 2p	- (*) Se acordă câte 2p. pentru fiecare aspect al cerinței (acces la cifrele unui număr pentru cele două valori, identificarea cifrelor comune distincte, formarea numărului cerut, tratare caz -1) (**) Se acordă punctajul pentru orice formă corectă de structură repetitivă sau decizională.
2	Pentru program corect - declarare a unei variabile care să memoreze un tablou - citirea datelor - memorare a numerelor conform cerinței(*) - afisarea rezultatelor - declarare a variabilelor simple, corectitudine globală a programului ¹⁾	10p. 1p. 1p. 6p. 1p. 1p	(*) Se acordă câte 2 p. pentru fiecare aspect - salvarea într-o variabilă auxiliară a fiecărei valori din vector -utilizarea unei structuri repetitive -determinari corecte
3	a) Pentru răspuns corect - coerență a descrierii algoritmului (*) - justificare a elementelor de eficiență b) Pentru program corect - operații cu fișiere: declarare, pregătire în vederea citirii, citire din fișier - determinare a valorilor cerute (*),(**) - utilizare a unui algoritm eficient (***) - declarare a variabilelor, afișare a datelor, corectitudine globală a programului ¹⁾	2p. 1p. 1p. 8p. 1p. 5p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar. O soluție posibilă utilizează un vector de frecvență a valorilor de două cifre care sunt prefixe. Se determina apoi numărul maxim de apariții. Ulterior, se face afișarea valorilor care au număr maxim de apariții în fișier. Se tratează cazul particular cerut în enunț prin utilizarea unui contor/comutator Complexitate O(n).

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.



Inspectoratul Școlar Județean
Iași



Informatică

Succes!



BACALAUREAT 2020

