Tema pentru perioada de Vacanță:   
Din variantele de bacalaureat **2009**, subiectul **3,** problema **3:** Variantele: **1;** **2;** **3;** **4;** **5;** **8;** **10**; **13**; **16**; **26**; **28**; **39**; **40**; **46**; **47**; **49**; **56**; **57**; **58**; **59**; **77**; **78**; **80**; **84**; **85**; **93**. Am vrut ca Tema să fie obligatorie.

Veți fi testați din probleme asemănătoare imediat ce reîncepe școala \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**1.** Fișierul text **bac.txt** conține, pe o singură linie, cel mult **1000** de numere naturale nenule cu cel mult **4** cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește de la tastatură un număr natural nenul **n** (**n**≤**999)** și numerele din fișierul **bac.txt** și care afișează pe ecran, separate prin câte un spațiu, toate numerele din fișier care sunt divizibile cu **n**. Dacă fișierul nu conține niciun astfel de număr, atunci se va afișa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul **bac.txt** conține numerele: **3 100 40** **70** **25** **5 80** **6 3798**, pentru **n**=**10** atunci pe ecran se va afișa: **100 40** **70** **80**

**2. Fiși**erul text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult **100** de numere întregi, fiecare număr având cel mult **4** cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele naturale nenule din fișier. Dacă nu există astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fișierul **NR.TXT** conține numerele: -**3** -**10** **0 7** -**5 7 51** -**800 6 3798**, atunci pe ecran se va afișa: **6 7 7 51** **3798**

**3. Fiși**erul text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult **100** de numere naturale, fiecare număr având cel mult **4** cifre. Scrieți un program C/C++ care citește toate numerele din fișier și afișează pe ecran, separate prin câte un spațiu, în ordine crescătoare, toate numerele din fișier care au cel puțin **3** cifre. Dacă fișierul nu conține astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**4. Fiși**erul text **NR.TXT** conține pe o singură linie, separate prin câte un singur spațiu, cel mult **100** de numere naturale, fiecare număr având cel mult **4** cifre. Scrieți un program C/C++ care citește numerele din fișierul **NR.TXT** și afișează pe ecran, separate prin câte un spațiu, în ordine descrescătoare, toate numerele din fișier care au cel mult **2** cifre. Dacă fișierul nu conține astfel de numere se va afișa pe ecran mesajul **NU EXISTA**.

**5. Scri**eți un program C/C++ care citește de la tastatură un număr natural **n** cu cel mult **8** cifre (**n**≥**10**) și care creează fișierul text **NR.TXT** ce conține numărul **n** și toate prefixele nenule ale acestuia, pe o singură linie, separate prin câte un spațiu, în ordine descrescătoare a valorii lor.

**Exemplu:** pentru **n**=**10305** fișierul **NR.TXT** va conține numerele: **10305 1030** **103 10** **1**

**6. Scr**ieți un program C/C++ care citește de la tastatură un număr natural **n** (**n**≤**50**) și care afișează pe prima linie a fișierului **BAC.TXT** primele **n** numere pare, pe a doua linie primele **n**-**1** numere pare, ..., pe linia **n** primul număr par. Numerele vor fi afișate pe fiecare linie în ordine crescătoare și vor fi separate prin câte un spațiu.

**0 2 4**

**0 2**

**0**

**Exemplu:** dacă **n**=**3** fișierul **BAC.TXT** va avea conținutul alăturat.

**7.** Scrieți un program C/C++ care citește de la tastatură un număr natural **n** (**n**≤**100)** și apoi cele **n** elemente, numere naturale cu cel mult **4** cifre fiecare, ale unui tablou unidimensional a. Programul determină și afișează pe prima linie a ecranului suma celor **n** elemente ale tabloului, pe a doua linie a ecranului suma primelor **n**-**1** elemente și așa mai departe astfel încât pe linia **n**-**1** se va afișa suma primelor două elemente, iar pe linia **n** primul element al tabloului.

**10**

**6**

**3**

**1**

**Exemplu:** dacă **n**=**4,** iar tabloul are elementele a=(**1,2,3,4)** programul va afișa valorile alăturate.

**8.** Scrieți un program C/C++, care citește de la tastatură un număr natural **n** (**n**≤**1000**) și afișează pe ecran, separați prin câte un spațiu, primii **n** termeni ai șirului:   
 **1,**  **2,1,** **3,2,1,** **4,3,2,1,** ...

construit astfel: prima grupă este formată din numărul **1,** a doua grupă este formată din numerele **2** și **1,** etc. Grupa a **k**-a, este formată din numerele **k**, **k**-**1,**..., **1.**

**Exemplu:** pentru **n**=**8** se vor afișa valorile **1 2 1 3 2 1 4 3.**

**11**

**21**

**30**

**40**

**51**

**16**

**17**

**10**

**1**

**9.** Fișierul text **BAC.TXT** conține mai multe numere naturale cu cel mult **6** cifre fiecare, câte un număr pe fiecare linie a fișierului.

Scrieți un program C/C++ care afișează pe ecran toate numerele din fișier, în aceeași ordine, câte cinci numere pe fiecare linie, separate prin câte un spațiu, cu excepția ultimei linii care poate conține mai puțin de cinci numere.

**Exemplu:** dacă fișierul are conținutul alăturat, pe ecran se afișează:

**11** **21** **30** **40** **51**

**16** **17** **10** **1**

**10.** Scrieți programul C/C++ care citește de la tastatură elementele unui tablou unidimensional cu exact **10** numere naturale, mai mici decât **1000**, determină și afișează pe ecran, separate prin câte un spațiu, numărul de elemente din șir care sunt multipli ai numărului **13** și, în continuare, pozițiile pe care acestea se găsesc în șir. Elementele tabloului sunt numerotate de la **1** la **10**.

**Exemplu:** dacă șirul citit este **2,** **6,** **26**, **14**, **130,** **11**, **8,** **23**, **39**, **52**, se vor afișa numerele **4 3 5 9 10**.

**11. Se** consideră un șir de **n** numere reale. Un număr **x** din acest șir se numește **median** dacă este egal cu media aritmetică a celorlalte **n**-**1** numere din șir.

**Exemplu:** elementul **x**=**5** este median pentru șirul **2 5 6 7** deoarece **5=**(**2+6+7)**/**3.**

Scrieți un program C/C++ care citește de la tastatură un număr natural **n** din intervalul [**2,50**] și apoi un șir de **n** numere reale și afișează pe ecran câte dintre elementele șirului sunt **mediane**.

**12.** Pe prima linie a fișierului **bac.in** se află un număr natural nenul **n** (**n**≤**1000**), iar pe a doua linie a fișierului se află un șir format din **n** numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult **4** cifre. Scrieți un program C/C++ care citește valorile din fișier și care afișează pe ecran mesajul **DA** dacă toate elementele pare din șir sunt în ordine crescătoare și mesajul **NU** în caz contrar.

**Exemplu:** dacă fișierul **bac.in** are conținutul alăturat, pe ecran se va afișa: **DA**

**7**

**10 113 12** **33** **42** **1354** **421**

**13.** Pe prima linie a fișierului **bac.in** se află un număr natural nenul **n** (**n**≤**1000**), iar pe a doua linie a fișierului se află un șir format din **n** numere naturale, despărțite prin câte un spațiu, fiecare număr fiind format din cel mult **4** cifre. Scrieți un program C/C++ care citește numerele din fișier și care afișează pe ecran mesajul **DA** dacă elementele pare în șir sunt în ordine crescătoare, iar cele impare sunt în ordine descrescătoare, și mesajul **NU** în caz contrar.

**Exemplu:** dacă fișierul **bac.in** are conținutul alăturat, pe ecran se va afișa: **DA**

**8**

**10 1133** **12** **331 42** **1354** **221 13**

**14.** Fișierul **bac.txt** conține pe prima linie un număr natural **n** (**n**≤**100)**, iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale nenule, cu cel mult **4** cifre fiecare. Scrieți programul C/C++ care citește de la tastatură un număr natural **k** (**k**≤**25**), construiește în memorie și afișează pe ecran un tablou unidimensional ce conține, în ordinea în care au fost citite, numerele de pe a doua linie a fișierului **bac.txt** care au cel puțin **k** divizori.

**Exemplu:** dacă **k**=**5,** iar fișierul are conținutul alăturat, atunci tabloul care se afișează este:

**6**

**100 9 400 56** **7 10**

**100 400 56**

**15.** Fișierul text **bac.txt** conține pe prima linie un număr natural **n** (**n**<**100)**, iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale, mai mici decât **30000** fiecare. Scrieți un program C/C++ care citește de la tastatură un număr natural **k** (**k**<**50**) precum și numerele din fișierul **bac.txt** și afișează pe ecran, cu câte un spațiu între ele, toate numerele de pe a doua linie a fișierului care sunt puteri ale lui **k**. Dacă nu există un asemenea număr pe a doua linie a fișierului, se afișează pe ecran mesajul **NU**. Un număr natural **x** este putere a lui **k** dacă există un număr natural **y** astfel încât **x**=**ky** .

**Exemplu:** dacă se citește de la tastatură **k**=**2,** iar fișierul are conținutul alăturat, atunci pe ecran se afișează numerele:

**8**

**32 56** **317 809 256 2 1 60**

**32** **256 2 1**

**16.** Scrieți un program C/C++ care citește de la tastatură un număr natural **n** ( **1** ≤**n** ≤**50**) și un șir de **n** numere naturale cu cel mult **4** cifre fiecare și care verifică dacă elementele șirului pot fi rearanjate astfel încât să respecte regula: al doilea element este cu **1** mai mare decât primul, al treilea cu **2** mai mare decât al doilea, ... , ultimul este cu **n**-**1** mai mare decât penultimul. Programul afișează pe ecran mesajul **DA** în caz afirmativ și mesajul **NU** în caz contrar.

**Exemplu:** pentru **n**=**4** și șirul **8,5,11**,**6** se afișează **DA** (elementele pot fi rearanjate astfel încât să respecte regula dată: **5,6,8,11**).

**17. F**ișierul **bac.in** conține pe prima linie un număr natural **n** (**0<n**<**5000**), iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale, formate din cel mult **4** cifre fiecare.

Scrieți un program C/C++ care determină și scrie în fișierul **bac.out**, toate numerele, citite de pe a doua linie a fișierului **bac.in**, care apar de cel puțin două ori. Numerele determinate se vor scrie în ordine crescătoare, pe aceeași linie, separate prin câte un spațiu.

**Exemplu:** dacă fișierul **bac.in** conține pe prima linie numărul **11**, iar pe linia a doua valorile **23** **12** **54** **12** **78** **345 67** **23** **78** **934 23** atunci fișierul **bac.out** va conține: **12** **23** **78**

**18. Fi**șierul text **numere.txt** conține, pe o singură linie, cel mult **1000** de numere naturale nenule de cel mult patru cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișierul **numere.txt**, creează fișierul text **pare.txt** și scrie pe prima lui linie, separate prin câte un spațiu, toate numerele pare citite, în ordinea în care acestea apar în fișierul **numere.txt**. Dacă fișierul **numere.txt** nu conține niciun număr par, atunci în fișierul **pare.txt** se va scrie mesajul nu exista.

**Exemplu:**

**numere.txt pare.txt**

**2 3 1 4 7 2 5 8 6 2 4 2 8 6**

**19. Fiș**ierul text **INTRARE.TXT** conține pe prima linie o valoare naturală **n** (**1<n**<**100)**, iar pe a doua linie, separate prin câte un spațiu, **n** numere naturale distincte, cu cel mult patru cifre.

Scrieți un program C/C++ care creează fișierul text **IESIRE.TXT** și scrie în el, pe prima linie, separate prin spațiu, toate valorile obținute ca sumă de două elemente distincte aflate pe linia a doua în fișierul **INTRARE.TXT**.

**Exemplu:** dacă fișierul **INTRARE.TXT** are următorul conținut:

**4**

**1 4 3 2**

atunci fișierul **IESIRE.TXT** va conține numerele:

**5 4 3 7 6 5** (nu neapărat în această ordine)

**20. Fi**șierul text **bac.txt** conține, pe o singură linie, cel puțin **2** și cel mult **100** de numere naturale nenule distincte de cel mult **4** cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește numerele din fișier și scrie pe ecran, în ordine crescătoare, separate prin spațiu, cele mai mici două numere dintre cele citite.

**Exemplu:** dacă fișierul **bac.txt** conține numerele: **1017** **48** **310 5710** **162** atunci se va afișa: **48** **162**

**21. Fiș**ierul text **bac.txt** conține, pe o singură linie, cel puțin **2** și cel mult **100** de numere naturale nenule distincte de cel mult **4** cifre fiecare, numerele fiind separate prin câte un spațiu. Scrieți un program C/C++ care citește toate numerele din fișier și scrie pe ecran ultima cifră a produsului celor mai mari două numere dintre cele citite.

**Exemplu:** dacă fișierul **bac.txt** conține numerele: **1017** **48** **312 5742** **162** atunci se va afișa: **4** (ultima cifră a produsului numerelor **1017**, **5742**)

**22. Fiș**ierul text **numere.in** are pe prima linie un număr natural **n** (**0<n**<**1000**) și pe următoarea linie **n** numere reale. Scrieți un program C/C++ care citește numerele din fișier și afișează pe ecran media aritmetică a numerelor aflate pe a doua linie a fișierului, care au partea întreagă formată din cel puțin două cifre.

**23. Se** citește de la tastatură un număr natural **n**, cu cel mult patru cifre, **n**≥**2.** Să se scrie în fișierul **prime.out**, pe aceeași linie, în ordine crescătoare, separate prin câte un spațiu, toate numerele prime mai mici sau egale cu **n**.

**Exemplu:** pentru **n**=**10**, în fișierul **prime.out** vor fi scrise numerele **2 3 5 7** .

**24. Scr**ieți programul C/C++ care citește de la tastatură un număr natural **n** (**1≤n**≤**100)**, un șir de câte **n** numere întregi, cu cel mult **5** cifre fiecare, notat **a1,a2,a3,…an**, apoi un al doilea șir de **n** numere întregi, cu cel mult **5** cifre fiecare, notat **b1,b2,b3,…bn**. Programul construiește în memorie și afișează pe ecran un șir **c** format din **n** numere calculate astfel: **ci=ai+bi**, pentru **i**=**1,2,3,**…**n**. Numerele afișate vor fi separate prin câte un spațiu.

**Exemplu:** pentru **n**=**4** și numerele **2,3,7,8** respectiv **43**,**3,1,8** se afișează **45** **6 8 16**.

**25. Scr**ieți programul C/C++ care citește de la tastatură un număr natural **n** (**1≤n**≤**100)**, un șir de câte **n** numere întregi, cu cel mult **5** cifre fiecare, notat **a1,a2,a3,…an**, apoi un al doilea șir de **n** numere întregi, cu cel mult **5** cifre fiecare, notat **b1,b2,b3,…bn**. Programul construiește în memorie și afișează pe ecran un șir C format din **n** numere calculate astfel: **ci=ai-bi**, pentru **i**=**1,2,3,**…**n**. Numerele afișate sunt separate prin câte un spațiu.

**Exemplu:** pentru **n**=**4** și numerele **2,3,7,8** respectiv **43**,**3,1,8** se afișează:

-**41** **0 6 0.**

**26. Scr**ieți programul C/C++ care citește de la tastatură un număr natural **n** (**1≤n**≤**99**), impar, și construiește în memorie un tablou unidimensional **A**=(**A1,** **A2,**…, **An**) cu elementele mulțimii {**1,2,**...,**n**} astfel încât elementele de pe poziții impare formează șirul crescător **1,2,**...,[(**n**+**1)**/**2]**, iar elementele de pe poziții pare șirul descrescător **n**,**n**-**1,**..., [(**n**+**1)**/**2]**+**1.**

**Exemplu:** pentru **n**=**11** se va construi tabloul **A** : 

Elementele tabloului se afișează pe ecran, separate prin câte un spațiu.