Subiecte C++/C ATESTAT 2011

1. Scrieți programul care citește un număr natural nenul **p** și afișează toate tripletele de numere naturale ce reprezintă laturile unor triunghiuri de perimetru **p**.
2. Scrieți programul care citește un număr natural nenul **n** și afișează numărul posibilităților de descompunere a numărului **n** în sumă de numere consecutive.
3. Scrieți programul care citește un număr natural nenul **n** cu cel mult **4** cifre și determină cel mai mic număr prim mai mare decât **n**.
4. Scrieți programul care citește un număr natural nenul **n** cu cel mult patru cifre și determină cel mai mare termen al șirului lui Fibonacci, mai mic sau cel mult egal cu **n**.
5. Se citeste un numar intreg strict pozitiv cu cel mult noua cifre. Sa se verifice daca cifrele lui sunt in ordine strict crescatoare.
6. Se citesc trei numere intregi strict positive. Sa se afiseze in ordine crescatoare. Sa se verifice daca ele formeaza sau nu o progresie geometrica.
7. Se citesc doua numere intregi strict pozitive. Sa se calculeze sis a se afiseze c.m.m.d.c al lor.
8. Se citeste de la tastatura un numar natural n (1<n<100) si un sir x de n valori intregi. Sa se construiasca un vector y care sa contina numai elementele pare din vectorul x.
9. Se citeste de la tastatura un numar natural n (1<n<100) si un sir x de n valori intregi. Verificati daca o valoare a, citita de la tastatura se gaseste in vector si daca da, afisti pozitiile pe care apare si numarul de aparitii.
10. Se citeste de la tastatura un numar natural n (1<n<100) si un sir x de n valori intregi. Sa se numere elementele din vectorul x care se gasesc in intervalul [a,b] cu a si b numere intregi citite de la tastatura, si sa se construiasca un alt vector cu cele care nu apartin intervalului.
11. Se citeste de la tastatura un numar natural n (1<n<100) si un sir x de n valori intregi. Sa se ordoneze crescator pana la o pozitie k citita de la tastatura si descrescator de la pozitia k la n.
12. Scrieţi un program C++ care citeşte de la tastatură un număr natural nenul n cu cel mult 3 cifre şi calculează suma:

.

Rezultatul se va afişa pe ecran cu exact trei zecimale

1. Fişierul text **bac.txt** conţine, pe o singură linie, cel mult **1000** de numere naturale nenule cu cel mult **4** cifre fiecare, numerele fiind separate prin câte un spaţiu. Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural nenul **n** (**n**≤**999**) şi numerele din fişierul **bac.txt** şi care afişează pe ecran, separate prin câte un spaţiu, toate numerele din fişier care sunt divizibile cu **n**. Dacă fişierul nu conţine niciun astfel de număr, atunci se va afişa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fişierul **bac.txt** conţine numerele: **3 100 40 70 25 5 80 6 3798**,

pentru **n=10** atunci pe ecran se va afişa: **100 40 70 80**

1. Fişierul text **NR.TXT** conţine pe o singură linie, separate prin câte un singur spaţiu, cel mult

**100** de numere **întregi**, fiecare număr având cel mult **4** cifre. Scrieţi un program **C/C++** care citeşte numerele din fişierul **NR.TXT** şi afişează pe ecran, separate prin câte un spaţiu, în ordine crescătoare, toate numerele **naturale nenule** din fişier. Dacă nu există astfel de numere se va afişa pe ecran mesajul **NU EXISTA**.

**Exemplu:** dacă fişierul **NR.TXT** conţine numerele: -**3 -10 0 7 -5 7 51 -800 6 3798**,

atunci pe ecran se va afişa: **6 7 7 51 3798**

1. Fişierului text **NR.TXT** conţine pe o singură linie, separate prin câte un singur spaţiu, cel mult **100** de numere naturale, fiecare număr având cel mult **4** cifre. Scrieţi un program **C/C++** care citeşte toate numerele din fişierul **NR.TXT** şi afişează pe ecran, separate prin câte un spaţiu, în ordine crescătoare, toate numerele din fişier care au cel puţin **3** cifre.

Dacă fişierul nu conţine astfel de numere se va afişa pe ecran mesajul **NU EXISTA**

1. Fişierul text **NR.TXT** conţine pe o singură linie, separate prin câte un singur spaţiu, cel mult **100** de numere naturale, fiecare număr având cel mult **4** cifre. Scrieţi un program **C/C++** care citeşte numerele din fişierul **NR.TXT** şi afişează pe ecran, separate prin câte un spaţiu, în ordine descrescătoare, toate numerele din fişier care au cel mult **2** cifre. Dacă fişierul nu conţine astfel de numere se va afişa pe ecran mesajul **NU EXISTA**.
2. Scrieţi un program **C/C++** care citeşte de la tastură un număr natural **n** cu cel mult **8** cifre (**n**≥**10**) şi care creează fişierul text **NR.TXT** ce conţine numărul **n** şi toate prefixele nenule ale acestuia, pe o singură linie, separate prin câte un spaţiu, în ordine descrescătoare a valorii lor.

**Exemplu:** pentru **n=10305** fişierul **NR.TXT** va conţine numerele:

**10305 1030 103 10 1**

1. Scrieţi un program C/C++ care citeşte de la tastatură două numere naturale nenule n şi m (2≤m≤10, 2≤n≤10) şi care construieşte în memorie şi apoi afişează o matrice A cu n linii (numerotate de la 1 la n) şi m coloane (numerotate de la 1 la m) cu proprietatea că fiecare element Aij memorează cea mai mică dintre valorile indicilor i şi j (1≤i≤n, 1≤j≤m).

Matricea se va afişa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu.

**Exemplu:** pentru n=4 şi m=5 se va afişa matricea alăturată.

1 1 1 1 1

1 2 2 2 2

1 2 3 3 3

1 2 3 4 4

1. Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2≤n≤24**) şi construieşte în memorie o matrice cu **n** linii şi **n** coloane ale cărei elemente vor primi valori după cum urmează:

- elementele aflate pe diagonala principală a matricei vor primi valoarea **0**

- elementele de pe prima coloană, cu excepţia celui aflat pe diagonala principală vor primi

valoarea **n**

- elementele de pe a doua coloană, cu excepţia celui aflat pe diagonala principală vor primi

valoarea **n-1**

...

- elementele de pe ultima coloană, cu excepţia celui aflat pe diagonala principală vor primi

valoarea **1**

Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii (ca în exemplu).

**Exemplu**: pentru **n=4** se va afişa matricea alăturată.

**0 3 2 1**

**4 0 2 1**

**4 3 0 1**

**4 3 2 0**

20. Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2≤n≤24**) şi construieşte în memorie o matrice cu **n** linii şi **n** coloane ale cărei elemente vor primi valori după cum urmează:

- elementele aflate pe diagonala secundară a matricei vor primi valoarea **0**

- elementele de pe prima linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea **n**

- elementele de pe a doua linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea **n-1**

**...**

- elementele de pe ultima linie, cu excepţia celui aflat pe diagonala secundară vor primi valoarea **1**

Programul va afişa matricea astfel construită pe ecran, câte o linie a matricei pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii (ca

în exemplu).

**Exemplu**: pentru **n=4** se va afişa matricea alăturată.

**4 4 4 0**

**3 3 0 3**

1. **0 2 2**

**0 1 1 1**

1. Scrieţi un program **C/C++** care citeşte de la tastatură două numere naturale **n** şi **m** (**2≤m≤10**, **2≤n≤10**) şi care construieşte în memorie şi apoi afişează o matrice **A** cu **n** linii (numerotate de la **1** la **n**) şi **m** coloane (numerotate de la **1** la **m**) cu proprietatea că fiecare element **Aij** memorează cea mai mare dintre valorile indicilor **i** şi **j** (**1≤i≤n**, **1≤j≤m**).

Matricea se va afişa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spaţiu.

**Exemplu:** pentru **n=4** şi **m=5** se va afişa matricea alăturată.

**1 2 3 4 5**

**2 2 3 4 5**

**3 3 3 4 5**

**4 4 4 4 5**

1. Scrieţi un program **C/C++** care citeşte de la tastatură două numere naturale **n** şi **p** (**2≤n≤20**, **1≤p≤20**) şi construieşte în memorie un tablou bidimensional cu **n** linii şi **p** coloane. Tabloul va fi construit astfel încât, parcurgând tabloul linie cu linie de sus în jos şi fiecare linie de la stânga la dreapta, să se obţină şirul primelor **n\*p pătrate perfecte impare**, ordonat strict crescător, ca în exemplu. Tabloul astfel construit va fi afişat pe ecran, fiecare linie a tabloului pe câte o linie a ecranului, cu câte un spaţiu între elementele fiecărei linii.

**Exemplu**: pentru **n=2**, **p=3** se va afişa tabloul alăturat:

**1 9 25**

**49 81 121**

1. Scrieţi un program **C/C++** care citeşte de la tastatură un număr natural **n** (**2<n<16**), construieşte în memorie şi afişează pe ecran o matrice cu **n** linii şi **n** coloane în care elementele de pe cele două diagonale sunt egale cu **0**, elementele care se află deasupra ambelor diagonale sunt egale cu **1**, elementele care se află sub ambele diagonale sunt egale cu **2**, iar restul elementelor sunt egale cu **3**.

Elementele matricei vor fi afişate pe ecran, câte o linie a matricei pe câte o linie a ecranului cu câte un spaţiu între elementele fiecărei linii.

**Exemplu**: pentru **n=5** se va afişa matricea alăturată. **(10p.)**

**0 1 1 1 0**

**3 0 1 0 3**

**3 3 0 3 3**

**3 0 2 0 3**

**0 2 2 2 0**

1. Se consideră un text alcătuit din cel mult **250** de caractere, în care cuvintele sunt formate doar din litere mici ale alfabetului englez şi sunt separate prin unul sau mai multe caractere **\***. Scrieţi un program **C/C++** care citeşte de la tastatură textul şi afişează pe ecran, pe câte o linie, toate secvenţele formate din câte două litere identice, ca în exemplu.

**Exemplu:** dacă textul citit este: se afişează

perechile alăturate.

**ii**

**ii**

**oo**

1. Scrieţi un program **C/C++** care citeşte de la tastatură două caractere **c1** şi **c2**, şi un text având cel mult **250** caractere (spaţii şi litere ale alfabetului englez), pe care îl modifică înlocuind toate apariţiile caracterului memorat în **c1** cu cel memorat în **c2** şi toate apariţiile caracterului memorat în **c2** cu cel memorat în **c1**. Programul afişează pe linii separate ale ecranului atât textul iniţial cât şi textul obţinut după efectuarea înlocuirilor.

**Exemplu:** dacă pentru **c1** se citeşte **a**, pentru **c2** se citeşte **o** iar textul citit este:

**hocus pocus preparatus**

se va afişa :

**hocus pocus preparatus**

**hacus pacus preporotus**

1. Scrieţi programul **C/C++** care citeşte de la tastatură un şir de cel mult **40** de caractere, format doar din litere mici ale alfabetului englez, şi care afişează pe ecran, pe o singură linie, toate vocalele ce apar în şirul citit. Vocalele vor fi afişate în ordinea apariţiei lor în şir, separate prin câte un spaţiu, ca în exemplu. Şirul citit conţine cel puţin o vocală şi se consideră ca fiind vocale următoarele litere: **a, e, i, o, u**.

**Exemplu**: dacă se citeşte şirul **calculator** atunci pe ecran se va afişa: **a u a o**

1. Să se scrie un program **C/C++** care citeşte de la tastatură un cuvânt format din cel mult **20** de caractere, doar litere mici ale alfabetului englez. Programul determină transformarea cuvântului citit prin înlocuirea fiecărei vocale a cuvântului, cu litera mare corespunzătoare, restul literelor nemodificându-se, ca în exemplu. Programul afişează pe ecran cuvântul obţinut, pe o singură linie. Se consideră vocale literele din mulţimea {**a,e,i,o,u**}.

**Exemplu**: pentru cuvântul **bacalaureat** se va afişa pe ecran: **bAcAlAUrEAt**

1. Se consideră un text format doar din spaţii şi litere mici ale alfabetului englez, care începe cu o literă şi care conţine cel puţin o vocală din multimea **{a,e,i,o,u}**. Scrieţi programul **C/C++** care citeşte de la tastatură un şir cu cel mult **100** de caractere, ca cel descris mai sus şi care determină transformarea acestuia prin înlocuirea fiecărei vocale din text cu litera imediat următoare din alfabet (**a** se înlocuieşte cu **b**, **e** se înlocuieşte cu **f** ş.a.m.d.). Programul va afişa pe ecran şirul obţinut.

**Exemplu:** dacă şirul citit este **examen de bacalaureat**, după modificare se afişează:

**fxbmfn df bbcblbvrfbt**

1. Subprogramul **nule** are **2** parametri: **a**, prin care primeşte un tablou unidimensional cu maximum **100** de numere întregi, cu cel mult **4** cifre fiecare şi **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile nule să se afle la sfârşitul tabloului. Ordinea în cadrul secvenţei de elemente nenule poate fi oricare. Tabloul modificat este furnizat tot prin parametrul **a**.

**Exemplu:** dacă **n=6**, **a=(12,0,0,-3,-8,0)**, după apel, acesta ar putea fi: **a=(12,-3,-8,0,0,0)**.

Scrieţi definiţia completă a subprogramului **nule**.

1. Subprogramul **aranjare** are **2** parametri: **a** prin care primeşte un tablou unidimensional cu maximum **100** de numere reale şi **n**, numărul de elemente din tablou. Subprogramul rearanjează elementele tabloului unidimensional astfel încât toate valorile negative să se afle pe primele poziţii, iar valorile pozitive în continuarea celor negative. Ordinea în cadrul secvenţei de elemente pozitive, respectiv în cadrul secvenţei de elemente negative, poate fi oricare. Tabloul modificat va fi furnizat tot prin intermediul parametrului **a**.

**Exemplu:** dacă tabloul are **6** elemente şi este de forma **(12, -7.5, 6.5, -3, -8,**

**7.5)**, după apel, acesta ar putea fi: **(-7.5, -3, -8, 12, 6.5, 7.5)**.

Scrieţi definiţia completă a subprogramului **aranjare**.

1. Scrieţi definiţia completă a subprogramului **count** care are **2** parametri prin care primeşte un tablou unidimensional cu maximum **100** de numere reale şi numărul de elemente din tablou. Subprogramul returnează numărul de elemente din tabloul unidimensional care sunt mai mari sau cel puţin egale cu media aritmetică a tuturor elementelor din tablou.

**Exemplu:** dacă tabloul are **6** elemente şi este de forma **(12, 7.5, 6.5, 3, 8.5, 7.5)**, subprogramul va returna valoarea **4**.

1. Scrieţi definiţia completă a subprogramului **interval** care are **2** parametri prin care primeşte un tablou unidimensional cu maximum **100** de numere naturale mai mici decât **1000** şi numărul de elemente din tabloul unidimensional. Subprogramul returnează numărul de elemente din tabloul unidimensional care aparţin intervalului închis determinat de primul şi respectiv ultimul element al tabloului.

**Exemplu:** dacă tabloul are **6** elemente şi este de forma **(12,27,6,8,9,2)**, subprogramul va returna valoarea **5**.

1. Scrieţi definiţia completă a subprogramului **multiplu** care are **3** parametri: **a**, prin care primeşte un tablou unidimensional cu maximum **100** de numere naturale mai mici decât **1000**, **n**, numărul de elemente ale tabloului şi **k**, un număr natural (**k**≤**9**). Subprogramul returnează numărul de elemente din tablou care sunt multipli ai numărului **k** şi au ultima cifră egală cu **k**.

**Exemplu:** dacă **n=6**, **a=(2,273,63,83,93,123)**, iar **k=3**, subprogramul va returna valoarea **4**.