

Subiectul II (30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera care corespunde răspunsului corect.

1. Se consideră o coadă în care inițial au fost introduse, în această ordine, elementele cu valorile 1 și 2:

1	2
---	---

. Se notează cu **AD(x)** operația prin care se adaugă elementul cu valoarea **x** în coadă și cu **EL** operația prin care se elimină un element din coadă. Câte elemente va conține coada în urma executării secvenței de operații: **AD(4); EL; EL; AD(5); EL; AD(3)**? **(4p.)**
- a. 3 b. 1 c. 2 d. 5
2. Câte grafuri neorientate, distincte, cu 4 vârfuri, se pot construi? Două grafuri se consideră distincte dacă matricele lor de adiacență sunt diferite. **(4p.)**
- a. 24 b. 4 c. 4⁶ d. 2⁶

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

3. În declararea alăturată, câmpurile **x** și **y** ale înregistrării pot memora coordonatele carteziene ale unui punct din planul **xOy**. Scrieți o secvență de instrucțiuni prin executarea căreia se memorează în variabila **C** coordonatele mijlocului segmentului cu capetele în punctele ale căror coordonate sunt memorate în variabilele **A** și **B**. **(6p.)**
- ```
struct punct
{
 float x,y;
}A,B,C;
```
4. Prin înălțimea unui arbore cu rădăcină înțelegem numărul de muchii ale celui mai lung lanț format din noduri distincte care are una dintre extremități în rădăcina arborelui. Scrieți care este înălțimea și care sunt frunzele arborelui descris prin următorul vector "de tați": **(6, 6, 5, 0, 6, 4, 4, 7)**. **(6p.)**
5. Scrieți un program **C/C++** care citește de la tastatură două numere naturale nenule **n** și **m** (**2 ≤ m ≤ 10**, **2 ≤ n ≤ 10**) și care construiește în memorie și apoi afișează o matrice **A** cu **n** linii (numerotate de la 1 la **n**) și **m** coloane (numerotate de la 1 la **m**) cu proprietatea că fiecare element **A<sub>ij</sub>** memorează cea mai mică dintre valorile indicilor **i** și **j** (**1 ≤ i ≤ n**, **1 ≤ j ≤ m**). Matricea se va afișa pe ecran, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru **n=4** și **m=5** se va afișa matricea alăturată.

**(10p.)**

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 2 | 2 | 2 | 2 |
| 1 | 2 | 3 | 3 | 3 |
| 1 | 2 | 3 | 4 | 4 |

**Subiectul III (30 de puncte)**

**Pentru itemul 1, scrieți pe foaia de examen litera corespunzătoare răspunsului corect.**

1. Utilizând metoda backtracking se generează în ordine lexicografică cuvintele de câte patru litere din mulțimea  $A=\{a, b, c, d, e\}$ , cuvinte care nu conțin două vocale alăturate. Primele opt cuvinte generate sunt, în ordine: **abab, abac, abad, abba, abbb, abbc, abbd, abbe**. Câte dintre cuvintele generate încep cu litera **b** și se termină cu litera **e**? (4p.)
- a. 9                                      b. 15                                      c. 12                                      d. 20

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

2. Pentru definiția de mai jos a subprogramului **f**, ce se afișează ca urmare a apelului **f(121,1);**? (6p.)
- |                                                                                                      |                                                                                                       |
|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
| <pre>//C void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { printf("%d",i); f(n/3,i+1); } }</pre> | <pre>//C++ void f(long n, int i) { if(n!=0)   if(n%3&gt;0)     { cout&lt;&lt;i; f(n/3,i+1); } }</pre> |
|------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------|
3. 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** ( $n \leq 999$ ) 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** (10p.)
4. Subprogramul **sub**, cu trei parametri, primește prin intermediul parametrilor:
- **v** un tablou unidimensional cu cel mult **100** de componente ce memorează numere întregi de cel mult **4** cifre fiecare;
  - **n** un număr natural nenul mai mic sau egal cu **100** ce reprezintă numărul efectiv de componente ale tabloului primit prin intermediul parametrului **v**;
  - **a** un număr întreg cu cel mult **4** cifre.
- Subprogramul **sub** returnează numărul componentelor tabloului primit prin intermediul parametrului **v** ale căror valori sunt strict mai mici decât valoarea parametrului **a**.  
**Exemplu:** pentru valorile **n=5**, **v=(1, 21, 9, 21, 403)**, **a=20** ale parametrilor, în urma apelului, subprogramul **sub** va returna valoarea **2**.
- a) Scrieți definiția completă a subprogramului **sub**. (4p.)
- b) Să se scrie un program **C/C++** care să citească de la tastatură un număr natural nenul **n** ( $n \leq 100$ ) și **n** numere întregi, fiecare având cel mult **4** cifre, și care, folosind apeluri utile ale subprogramului **sub**, să afișeze pe ecran mesajul **DA** dacă oricare două dintre cele **n** numere întregi citite sunt distincte două câte două, sau mesajul **NU** în caz contrar.  
**Exemplu:** pentru **n=6** și cele **n** numere citite de la tastatură: **47 183 69 8 134 -56** se va afișa pe ecran mesajul **DA** (6p.)