



**Subiectul II (30 de puncte)**

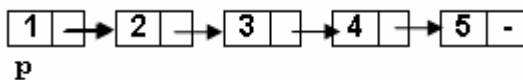
Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera care corespunde răspunsului corect.

1. În declararea alăturată, variabila **p** memorează în câmpul **x** abscisa, iar în câmpul **y** ordonata unui punct din planul **xOy**. Dacă punctul se află în interiorul suprafeței dreptunghiulare determinate de punctele **A(1,1)**, **B(4,1)**, **C(4,3)**, **D(1,3)**, care dintre expresiile de mai jos are valoarea **1**? (4p.)
- ```
struct  
{float x;  
float y;}p;
```
- a. **(p.x>1)&&(p.x<4)&&(p.y>1)&&(p.y<3)**  
b. **(x.p>1)&&(x.p<4)&&(y.p>1)&&(y.p<3)**  
c. **(p.x>1)&&(p.x<4) || (p.y>1)&&(p.y<3)**  
d. **(p(x)>1)&&(p(x)<4) || (p(y)>1)&&(p(y)<3)**
2. Ce valoare are variabila **s** de tip șir de caractere după executarea instrucțiunilor de mai jos?  
**strncpy(s, strstr("informatica", "form"), strlen("BAC009"));**  
**s[6]='\0';** (4p.)
- a. **format**      b. **informat**      c. **inform**      d. **informBAC**

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

3. Determinați ultima valoare (notată cu „?”) din vectorului „de tați” (**0, 1, 1, 2, 3, 3, ?**) astfel încât arborele cu rădăcină, cu **7** noduri, numerotate de la **1** la **7**, descris de acest vector, să aibă pe fiecare nivel **n** exact **2<sup>n</sup>** noduri, nodul rădăcină fiind pe nivelul **n=0**, și fiecare nod să aibă cel mult doi descendenți. Scrieți matricea de adiacență a unui arbore astfel definit. (6p.)

4. Fiecare element al unei liste simplu înlănțuite, alocată dinamic, reține în câmpul **nr** un număr întreg, iar în câmpul **urm** adresa următorului element din listă sau **NULL** dacă nu există un element următor. Ce valori au variabilele întregi **a** și **b** după executarea secvenței alăturate, dacă variabila **p** reține adresa primului element al listei de mai jos, iar variabila **q** este de același tip cu **p**? (6p.)
- ```
q=p;  
a=p->urm->nr;  
while (q->urm!=NULL)  
{  
    q->urm->nr=2*q->nr+1;  
    q=q->urm;  
}  
b=q->nr;
```



5. Scrieți un program **C/C++** care citește de la tastatură un număr natural **n** (**1 ≤ n ≤ 6**) apoi construiește în memorie o matrice cu **n** linii și **n** coloane, astfel încât parcurgând liniile matricei de sus în jos și de la stânga la dreapta se obțin, în prima linie primele **n** numere ale șirului Fibonacci în ordine **crescătoare**, în linia a doua următoarele **n** numere ale șirului Fibonacci în ordine **descrescătoare**, în linia a treia următoarele **n** numere ale acestui șir în ordine **crescătoare**, și așa mai departe, ca în exemplu. Elementele șirului Fibonacci se obțin astfel: primul element este **0**, al doilea este **1**, iar elementele următoare se obțin însumând cele două elemente care preced elementul curent. Astfel, primele **16** elemente ale acestui șir sunt: **0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610**.

Programul afișează pe ecran matricea obținută, câte o linie a matricei pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

**Exemplu:** pentru **n=4** se obține matricea alăturată.

|            |            |            |            |
|------------|------------|------------|------------|
| <b>0</b>   | <b>1</b>   | <b>1</b>   | <b>2</b>   |
| <b>13</b>  | <b>8</b>   | <b>5</b>   | <b>3</b>   |
| <b>21</b>  | <b>34</b>  | <b>55</b>  | <b>89</b>  |
| <b>610</b> | <b>377</b> | <b>233</b> | <b>144</b> |

(10p.)

