

**Subiectul III (30 de puncte)**

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

1. Utilizăm metoda backtracking pentru generarea tuturor modalităților de a scrie numărul 6 ca sumă a cel puțin două numere naturale nenule. Termenii fiecărei sume sunt în ordine crescătoare. Soluțiile se generează în ordinea: 1+1+1+1+1+1, 1+1+1+1+2, 1+1+1+3, 1+1+4, 1+2+3, 1+5, 2+2+2, 2+4 și 3+3. Se aplică exact aceeași metodă pentru scrierea lui 9. Câte soluții de forma 2+... vor fi generate? **(4p.)**
- a. 4                                      b. 3                                      c. 2                                      d. 5

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

2. În secvența alăturată variabilele *i*, *j* și *aux* sunt întregi, iar *v* memorează un tablou unidimensional ale cărui elemente au următoarele valori: *v*[1]=2, *v*[2]=13, *v*[3]=1, *v*[4]=5, *v*[5]=-4. Care este numărul de interschimbări care se efectuează în urma executării secvenței alăturate? Dați un exemplu de alte 5 valori pentru elementele tabloului astfel încât, în urma executării secvenței alăturate, să nu se efectueze nicio interschimbare. **(6p.)**
- ```
for(i=1;i<5;i++)
for(j=5;j>i;j--)
if(v[i]<v[j])
{ aux=v[i];
v[i]=v[j];
v[j]=aux;
}
```
3. Pe prima linie a fișierului **bac.in** se află un număr natural nenul *n* ( $n \leq 1000$ ), 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
- |   |
|---|
| 8 |
|---|
- alăturat, pe ecran se va afișa: **DA**                                      **(10p.)** 10 1133 12 331 42 1354 221 13
4. Considerăm definite subprogramele:
- **pr**, care primește prin intermediul parametrului *x* un număr natural cu cel mult 4 cifre și returnează 1 dacă numărul este prim și 0 în caz contrar;
  - **sdiv**, care primește prin intermediul parametrului *y* un număr natural cu cel mult 4 cifre și returnează suma tuturor divizorilor numărului *y*.
- a) Scrieți numai antetul subprogramelelor **pr** și **sdiv**. **(4p.)**
- b) Scrieți un program C/C++ care citește de la tastatură un număr natural *n* ( $n < 1000$ ) și care afișează pe ecran toate numerele mai mici decât *n* cu proprietatea că suma divizorilor lor este număr prim. Se vor utiliza apeluri utile ale subprogramelelor **pr** și **sdiv**.
- Exemplu:** dacă *n*=20, atunci programul va afișa: 2 4 9 16. **(6p.)**