

Subiect de la seria de Mate

Subiectul 1. (1 punct)

Ce afișează instrucțiunea de mai jos? Justificați.

```
printf("%d %d\n", (11 && 22), (11 & 22));
```

Subiectul 2. (1 punct)

Funcția *verificaPrim* de mai jos este scrisă pentru a verifica dacă un număr natural n este prim. Funcția trebuie să returneze 1 dacă n este prim sau 0 altfel. Funcția *verificaPrim* de mai jos este scrisă greșit.

```
int verificaPrim(int n)
{
    int i;
    if (n <= 1)
        return 1;
    if (n % 2 == 0)
        return 0;
    for (i = 3; i < n; i = i + 2)
        if (n % i == 0)
            return 0;
    return 1;
}
```

- (a) dați exemplu de o intrare n pentru care funcția scrisă returnează corect rezultatul. Justificați. (0.25 puncte)
- (b) dați exemplu de o intrare n pentru care funcția scrisă returnează incorect rezultatul. Justificați. (0.25 puncte)
- (c) scrieți varianta corectă a funcției *verificaPrim* (puteți porni de la actuala variantă sau puteți rescrie în totalitate funcția). (0.5 puncte)

Subiectul 3. (1 punct)

Ordonăți crescător funcțiile de mai jos pe baza creșterii lor asimptotice, precizând de fiecare dată funcțiile cu același ordin de creștere:

$$f_1(n) = 2^n, f_2(n) = n^3 + 3, f_3(n) = \sqrt{n}, f_4(n) = 3^n + 5, f_5(n) = 4^n$$

$$f_6(n) = 10 \cdot \log_2(n), f_7(n) = n \cdot \sqrt{n}, f_8(n) = 2 \cdot n \cdot \log_2(n), f_9(n) = \ln(n), f_{10}(n) = 4 \cdot n + 5.$$

Subiect de la seria de Mate

Subiectul 4. (1 punct)

```
#include <stdio.h>
#include <stdlib.h>
int f1(int a) {return 5*a;}
int f2(int b) {return b-2;}
int f3(int *c) {return f2(f1(*c));}
void main()
{
    int a=0,b=1,c=2;
    printf("%d %d %d \n", f1(b), f2(c), f3(&a));
}
```

Analizați programul de alături. Dacă programul este corect spuneți ce va afișa argumentând răspunsul. Dacă programul nu este corect explicați de unde provine eroarea.

Subiectul 5. (2 puncte)

Scrieți o funcție care primește ca parametri numele unui tablou unidimensional de numere întregi și numărul de elemente al acestuia și returnează suma maximă a unei secvențe (șir de numere consecutive) formate doar din valori strict pozitive ale tabloului primit ca parametru. Precizați și argumentați complexitatea funcției.

Exemplu: Pentru vectorul (1, 2, 3, -1, 0, 4, 5, -2) suma maximă a unei secvențe este 9.

Subiect de la seria de Mate

Subiectul 6. (2 puncte)

Scrieți o funcție care primește ca parametru un număr natural nenul n și returnează un tablou bidimensional pătratic, alocat dinamic, având următoarea formă:

pentru $n = 3$

0	0	0
0	1	1
0	1	2

pentru $n = 4$

0	0	0	0
0	1	1	1
0	1	2	2
0	1	2	3

Subiectul 7. (2 puncte)

Fișierul *cuvinte.in* conține pe prima linie un număr natural n nenul iar pe următoarele linii un text în care cuvintele sunt despărțite prin spații. Realizați un program care să scrie în fișierul text *cuvinte.out* toate cuvintele de lungime n din fișierul *cuvinte.in* sau mesajul "Imposibil" dacă în fișierul de intrare nu există niciun cuvânt de lungime n .

Subiect de la seria de Info

Subiectul nr. 1 (2 puncte)

- a) Scrieți o funcție care să citească numărul de elemente ale unui tablou unidimensional, să aloce dinamic tabloul respectiv și apoi să citească valorile elementelor sale. (0.5 puncte)
- b) Scrieți o funcție care să afișeze elementele unui tablou unidimensional de numere reale. (0.5 puncte)
- c) Scrieți o funcție care să calculeze poziția pe care se află valoarea minimă a unei secvențe cuprinse între doi indici i și j ($0 \leq i \leq j < n$) dintr-un tablou unidimensional format din n numere reale. (0.5 puncte)
- d) Scrieți un program care, folosind apeluri utile ale funcțiilor definite anterior, citește de la tastatură un tablou unidimensional format din n numere reale, îl sortează și apoi îl afișează. Precizați și argumentați complexitatea programului obținut. (0.5 puncte)

Observație: Nu este permisă utilizarea unor variabile globale!

Subiectul nr. 2 (1 punct)

Scrieți o funcție care primește ca parametru un număr natural nenul n și returnează un tablou bidimensional pătratic, alocat dinamic, având următoarea formă (pentru $n = 4$):

0	0	0	0
0	1	1	1
0	1	2	2
0	1	2	3

Subiect de la seria de Info

Subiectul nr. 3 (2 puncte)

Fișierul *cuvinte.in* conține pe prima linie un număr natural nenul n , iar pe următoarele linii un text în care cuvintele sunt despărțite prin spații și semnele de punctuație uzuale. Scrieți un program care să scrie în fișierul text *cuvinte.out* toate cuvintele de lungime n din fișierul *cuvinte.in* sau mesajul "Imposibil!" dacă în fișierul de intrare nu există nici un cuvânt cu proprietatea cerută.

Subiectul nr. 4 (2 puncte)

a) Scrieți o funcție generică de căutare cu următorul antet:

`void * cautare(const void * x, const void * t, int n, int d, int (* cmpValori)(const void *, const void *))`

Funcția trebuie să returneze un pointer generic către prima apariție a valorii x în tabloul unidimensional t format din n elemente, fiecare având dimensiunea d octeți sau pointerul *NULL* dacă valoarea x nu se găsește în tablou.

Funcția comparator *cmpValori* se consideră că returnează 1 dacă valorile aflate la adresele primite ca parametrii sunt egale sau 0 în caz contrar. (1 punct)

b) Scrieți un program care citește de la tastatură un număr întreg r , un tablou unidimensional v format din n numere întregi și afișează, folosind apeluri utile ale funcției *cautare*, toate pozițiile pe care apare numărul x în tabloul v sau mesajul "Valoare inexistentă!" dacă numărul x nu se găsește în tablou. (1 punct)

Subiect de la seria de Info

Subiectul nr. 5 (2 puncte)

- a) Definiți o structură *Student* care să permită memorarea numelui, notelor, mediei și grupei corespunzătoare unui student. Scrieți o funcție care să calculeze mediile celor n studenți ale căror date sunt memorate într-un tablou unidimensional t cu elemente de tip *Student*. (1 punct)
- b) Folosind funcția *qsort* din biblioteca *stdlib.h*, sortați elementele unui tablou unidimensional t format din n elemente de tip *Student* în ordinea descrescătoare a mediilor, iar în cazul unor medii egale studenții respectivi se vor ordona alfabetic. Implementați funcția comparator corespunzătoare și scrieți apelul funcției *qsort*. (0.5 puncte)
- c) Scrieți o funcție care să scrie într-un fișier binar informațiile despre studenții dintr-o grupă. Funcția va avea ca parametrii un tablou unidimensional t cu n elemente de tip *Student* și un număr natural nenul g reprezentând numărul unei grupe. Numele fișierului binar va fi *grupa_g.bin*. (0.5 puncte)