

Simulare județeană - Examenul național de bacalaureat, decembrie 2024

Proba E. d)
Informatică
Limbajul C/C++

Varianta 1

Filieră teoretică, profil real, specializare matematică-informatică / matematică-informatică intensiv informatică
Filieră vocațională, profil militar, specializare matematică-informatică

SUBIECTUL I

(20 de puncte)

Pentru fiecare dintre itemii de la 1 la 5, scrieți pe foaia de examen litera corespunzătoare răspunsului corect. Fiecare răspuns corect se notează cu 4 puncte.

- Care dintre expresiile C/C++ de mai jos este echivalentă cu expresia următoare $!((a < 2025) \&\& (b > 2024))$?
 a. $(a > 2024) \&\& (b < 2025)$ b. $!(a < 2024) \parallel !(b > 2025)$
 c. $!(a < 2024) \&\& !(b > 2025)$ d. $!(a < 2024) \&\& !(b > 2025)$
- Utilizând metoda backtracking se generează în ordine crescătoare toate numerele de patru cifre formate numai cu ajutorul cifrelor 4, 5, 2, astfel încât suma a oricăror două cifre vecine să fie un număr par, atunci cel de-al șaptelea număr scris este:
 a. 2424 b. 2442 c. 2422 d. 2444
- Se consideră subprogramul alăturat:
 Indicați ce se afișează în urma apelului de mai jos:
Info(2)
 a. 1 8 4 5 b. 5 8 1 4
 c. 1 4 8 5 d. 8 5 4 1


```
void Info(int x)
{
    if(x <= 6)
    {
        cout << x-1 << ' ';
        Info(x+3);
        cout << x+3 << ' ';
    }
}
```
- Se consideră arborele dat prin următorul vector de "tați" (4, 5, 5, 3, 0, 3, 2, 2, 2, 1). Indicați care este lungimea celui mai lung lanț elementar care unește două frunze ale arborelui.
 a. 4 b. 7 c. 3 d. 6
- Se consideră graful neorientat definit prin mulțimea vârfurilor {1, 2, 3, 4, 5, 6} și mulțimea muchiilor {[1,6], [2,3], [2,4], [3,5], [5,6]}
 Indicați numărul de muchii ce trebuie adăugate pentru ca acest graf să devină complet
 a. 2 b. 10 c. 3 d. 12

SUBIECTUL al II-lea

(40 de puncte)

- Se consideră algoritmul alăturat descris în pseudocod:
 S-a notat cu $x \% y$ restul împărțirii numărului natural x la numărul natural nenul y și cu $[z]$ partea întreagă a numărului real z .
 a. Scrieți numărul afișat dacă pentru variabila **n** se citește valoarea **74589**. (6p.)
 b. Scrieți toate numerele de exact trei cifre care pot fi citite pentru variabila **n** astfel încât, pentru fiecare dintre acestea, numărul afișat în urma executării algoritmului să fie **24**. (6p.)
 c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
 d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască structura **repetă...până când** cu o structură repetitivă de alt tip. (6p.)

citește n (număr natural nenul)
 $m \leftarrow 0$
repetă
 $c \leftarrow n \% 10$;
 $n \leftarrow [n/10]$
 dacă $c > 5$ **atunci**
 $c \leftarrow [c/2]$
 ■
 $m \leftarrow m * 10 + c$
până când $n = 0$
 scrie m

- Variabila **a** memorează simultan date specifice despre produsele alimentare aflate într-un depozit astfel: denumire produs (șir de caractere de cel mult 30 de caractere), preț, cantitate (numere naturale nenule), firma producătoare (șir de caractere de cel mult 20 de caractere) și data la care expiră produsul. Știind că în depozit se află maxim 300 de produse în stoc și că expresiile C/C++ de mai jos au ca valori datele referitoare la primul produs din stoc, scrieți în C/C++ definiția unei structuri cu eticheta **alimente** care permite memorarea datelor de mai sus și declarați corespunzător variabila **a**.
a[0].denumire, a[0].pret, a[0].cantitate, a[0].firmă, a[0].de.an, a[0].de.luna, a[0].de.zi (6p.)

- În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg iar variabila **A** este de tip tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 0 la 5, cu elemente numere întregi. Înainte de executarea

secvenței toate elementele tabloului sunt nule. Fără a utiliza alte variabile, scrieți o instrucțiune care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, tabloul memorat în variabila **A** să aibă elementele din figura de mai jos.

```
for(i=0;i<6;i++)
  for(j=0;j<6;j++)
    .....
```

(6p.)

0	1	2	3	4	5
1	0	1	2	3	4
2	1	0	1	2	3
3	2	1	0	1	2
4	3	2	1	0	1
5	4	3	2	1	0

SUBIECTUL al III-lea

(30 de puncte)

1. Se consideră subprogramul **nrdivimp**, cu parametrii **a** și **b**, prin care primește două valori naturale nenule $a \leq b < 10^9$. Subprogramul afișează, separate prin câte un spațiu, în ordine descrescătoare, toate numerele cu număr impar de divizori din intervalul **[a, b]** sau -1 dacă nu există astfel de numere. Scrieți definiția completă a subprogramului.

Exemplu: dacă **a=10** și **b=100** subprogramul va afișa **100 81 64 49 36 25 16**

(10p.)

2. Un șir cu maximum 255 de caractere conține cuvinte separate prin unul sau mai multe spații. Cuvintele sunt formate numai din litere mici ale alfabetului englez. Scrieți un program C/C++ care citește un astfel de șir și îl afișează modificat astfel: prima și ultima literă a fiecărui cuvânt fiind afișată ca literă mare.

(10p.)

Exemplu: pentru șirul: **azi avem proba la informatica**
se va afișa textul: **AzI AveM ProBA LA InformaticA**

3. Fișierul **numere.in** memorează cel puțin 3 numere și cel mult **10000** de numere naturale cu cel mult patru cifre reprezentând înălțimile vârfurilor munților Carpații din România. Se cere să se afișeze pe ecran, în ordine crescătoare, înălțimile celor mai înalte două vârfuri separate prin câte un spațiu.

Pentru determinarea vârfurilor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare.

Exemplu: dacă fișierul **numere.in** conține numerele **2520 1659 2044 2136 1959 1878 2563 1849 2503**, se afișează pe ecran **2520 2563**.

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia.

(2p.)

b) Scrieți programul C/C++ corespunzător algoritmului descris.

(8p.)