



Examenul național de bacalaureat 2022

Proba E.d)

Informatică

Limbajul C/C++

Testul 1

Filiera teoretică, profil real, specializarea matematică-informatică/matematică-informatică intensiv informatică

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de 3 ore.
- Identificatorii utilizați în rezolvări trebuie să respecte precizările din enunț (bold), iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora (eventual în formă prescurtată). Datele de intrare se consideră corecte, validarea lor nefiind necesară.
- În grafurile din cerințe oricare arc/muchie are extremități distincte și oricare două arce/muchii diferă prin cel puțin una dintre extremități.

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.

1. Variabilele **a, b** sunt de tip întreg și pot memora numere naturale. Expresia C/C++ alăturată $(a+b)\%2==1$ este echivalentă cu expresia:

- a. $(a\%2 != 1 \ \&\& \ b\%2 != 0) \ || \ (a\%2 != 0 \ \&\& \ b\%2 != 1)$
- b. $(a\%2 != 1 \ \&\& \ b\%2 != 0) \ \&\& \ (a\%2 != 0 \ \&\& \ b\%2 != 1)$
- c. $(a\%2 != 1 \ || \ b\%2 != 0) \ || \ (a\%2 != 0 \ || \ b\%2 != 1)$
- d. $(a\%2 != 0 \ \&\& \ b\%2 != 0) \ || \ (a\%2 == 0 \ \&\& \ b\%2 == 0)$

2. Subprogramul **f** este definit alăturat. Scrieți ce se afișează în urma apelului de mai jos.
f(124);

```
void f(int n)
{
    if(n>0)
        {if(n%2==0)
            {f(n/10);
             cout<<n%10<<' '; | printf("%d ",n%10);
             f(n/10);
            }
          else
            {f(n-1);
             cout<<n%10<<' '; | printf("%d ",n%10);
            }
        }
}
```

- a. 1 2 4 1 2 4
- b. 4 2 1 4 2 1

- c. 1 2 1 4 1 2 1
- d. 1 2 1 1 2 1



3. În declararea alăturată, câmpurile **den** și **nr_loc** ale înregistrării reprezintă denumirea, respectiv numărul de locuitori ale unei țări. Indicați o expresie C/C++ care are valoarea **1** dacă și numai dacă țara memorată în variabila **t** începe cu **R** și are numărul de locuitori cel puțin **1000000000**.

struct tara

{char den[20];int nr_loc;};

a. t.nrloc>=1000000000 && t.den[0]=='R';

b. t.den[1]=='R' && t.nrloc>=1000000000;

c. nrloc.t>=1000000000 && den[0].t=='R';

d. t.nrloc>=1000000000 && t.den[0]=='R';

4. Se consideră un graf neorientat **conex, aciclic**. Dacă graful are **10** noduri, atunci numărul maxim de componente conexe care se pot obține prin mutarea muchiilor este:

a. 6

b. 7

c. 5

d. 4

5. Numărul de muchii care trebuie adăugate la un arbore cu **11** noduri pentru a deveni un graf neorientat complet este:

a. 55

b. 45

c. 65

d. 44

Subiectul II(40 de puncte)

1. Se consideră algoritmul alăturat, descris în pseudocod.

S-a notat cu **a%b** restul împărțirii numărului natural **a** la numărul natural nenul **b**, și cu **[a]** partea întreagă a numărului real **a**.

citește n

(număr natural nenul)

d←2;p←0;c←0

repetă

cât timp n%d=0 execută

n←[n/d]

p←p+1

c←c+1

■

p←p+1

d←d+1

până când n=1

dacă c=d-1 atunci scrie 1

altfel scrie d

■



- a) Scrieți ce se afișează dacă se citește pentru numărul **n** valoarea **2020**. (6p.)
- b) Scrieți cel mai mic număr și cel mai mare număr din intervalul **[10,10000]** care poate fi citit pentru variabila naturală **n** astfel încât, în urma executării algoritmului, valoarea afișată să fie **1** (6p.)
- c) Scrieți programul **C/C++** corespunzător algoritmului dat. (10p.)
- d) Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind adecvat structura **repetă...până când** cu o structură repetitivă de alt tip. (6p.)

2. Variabilele **i**, **nr** sunt întregi, iar variabila **p** memorează, pentru fiecare dintre cele **15** puncte din sistemul **XoY**, abscisa și ordonata fiecărui punct.

Fără a utiliza alte variabile decât cele menționate, scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze mesajul **da** dacă toate punctele aparțin dreptunghiului care are diagonala determinată de punctele **(3,2)** și **(9,10)**, sau mesajul **nu** în caz contrar.

struct punct

{int abs;

int ord;

}p[15];

(6p.)

3. Un text are cel mult **100** de caractere, iar cuvintele sale sunt formate numai din litere mari ale alfabetului englez și sunt separate prin câte un spațiu. Scrieți un program **C/C++** care citește de la tastatură un text de tipul precizat mai sus și îl transformă în memorie prin eliminarea cuvintelor care conțin subșirurile **TIC**, **ICT** sau **IT** sau pot fi reduse la aceste subșiruri prin eliminarea unor caractere. Programul afișează pe ecran textul obținut.

Exemplu: pentru textul: **MATERIILE DE BACALAUREAT POT FI ROMANA MATEMATICA SI INFORMATICA**

se afișează: **MATERIILE DE BACALAUREAT POT FI ROMANA SI**

Exemplu: pentru textul: **CEILALȚI AU DREPTATE** se afișează: **AU DREPTATE** (6p.)

Subiectul III (30 de puncte)

1. Un număr **k** este deosebit dacă există un număr natural **p** astfel încât **k=p+suma(p)**, unde **suma(p)** este suma cifrelor lui **p**. Spre exemplu, **k=21** este număr deosebit (**p=15** și **k=15+6**).

Subprogramul *perfecte*, are doi parametri:

- **n**, prin care primește un număr natural (**1≤n<50**)
- **v**, prin care furnizează primele **n** numere naturale deosebite într-un tablou unidimensional cu **n** elemente.

Scrieți definiția completă a subprogramului.

Exemplu: dacă **n=7** atunci după apel subprogramul returnează tabloul **v=(2,4,6,8,10,11,12)**. (10p.)

2. Suprafața unui oraș este împărțită în **n*n** zone de formă pătrată dispuse pe **n** rânduri și **n** coloane, numerotate de la **1** la **n**, fiecare zonă fiind destinată pentru a fi zonă verde, fie pentru altceva. Zonelor li se atribuie câte un cod, în funcție de destinație: codul **0** pentru zonă verde și codul **1** pentru altceva.

Orașul este străbătut de la nord la sud și respectiv de la vest la est de două bulevarde drepte, care împart orașul în patru sectoare. Zona bulevardelor este codificată cu **2**.



Scrieți un program C/C++ care citește de la tastatură un număr natural, n ($n \in [1, 102]$), coloana c , de unde pornește bulevardul de la nord la sud, linia l de unde pornește bulevardul de la vest la est și cele $n \times n$ elemente ale unui tablou bidimensional, reprezentând codurile corespunzătoare zonelor verzi sau de altă destinație din oraș, în ordinea parcurgerii lor rând pe rând, și a fiecărui rând de la stânga la dreapta, în această ordine.

Programul determină numărul maxim de zone verzi dintre cele patru sectoare formate de cele două bulevarde. Dacă nu există nici o astfel de zonă verde în oraș se afișează pe ecran mesajul **nu exista**.

Exemplu: pentru $n=8$, $c=4$, $l=3$ și tabloul alăturat se afișează pe ecran numărul 8. (10p.)

0	0	1	2	1	1	0	1
1	0	1	2	0	1	1	0
2	2	2	2	2	2	2	2
0	0	1	2	1	1	1	1
0	0	1	2	0	0	0	0
0	0	1	2	1	1	1	1
1	1	1	2	0	0	1	0
0	0	1	2	1	1	1	1

3. Numim **secvență primă** a unui șir de numere naturale un subșir al acestuia, format din termeni aflați pe poziții consecutive în șirul dat, subșir care începe și se termină cu aceeași valoare primă. Lungimea secvenței este egală cu numărul de termeni ai acesteia. Fișierul **bac.txt** conține un șir de cel puțin două și cel mult 10^6 numere naturale din intervalul $[10, 99]$. Numerele sunt separate prin câte un spațiu. În șir există cel puțin doi termeni primi egali. Se cere să se determine secvențele prime din acest șir care au lungime maximă și să se afișeze pe prima linie a ecranului lungimea maximă determinată, iar pe următoarea linie, pentru fiecare astfel de secvență, valoarea primului său termen prim. Numerele de pe a doua linie sunt afișate în ordine crescătoare, separate prin câte un spațiu.

Proiectați un algoritm eficient din punct de vedere al timpului de execuție.

Exemplu: dacă fișierul **bac.txt** are conținutul:

11 12 17 15 13 10 23 41 55 55 12 13 33 23 31 10 99

atunci se afișează pe ecran valorile:

8

13 23

- Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. (2p.)
- Scrieți programul C/C++ corespunzător algoritmului descris. (8p.)