

Examenul național de bacalaureat – decembrie 2022

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

Simulare județeană

Filieră teoretică, profil real, specializare științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă zece puncte din oficiu.
- Timpul de lucru efectiv este de trei 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 muchie are extremități distincte și oricare două 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.

- Variabilele x și y sunt tip întreg. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă valorile întregi nenule memorate în variabilele x și y sunt impare.
 - $(x\%2==y\%2) \ \&\& \ (x*y>0)$
 - $(x\%2!=0) \ || \ (y\%2!=0)$
 - $!(x\%2==0) \ || \ !(y\%2==0)$
 - $!(x\%2==0) \ \&\& \ !(y\%2==0)$
- În secvența de mai jos, toate variabilele sunt de tip întreg.

```
for(i=0;i<5;i++){
    for(j=0;j<5;j++){
        if(.....)cout<<1<<' ';
        else cout<<0<<' ';
    }
    cout<<endl;
}
```

1	0	1	0	1
0	0	0	0	0
1	0	1	0	1
0	0	0	0	0
1	0	1	0	1

Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran cifrele 1 și 0 din figura de mai sus, în această ordine.

 - $i\%2==0\ \&\& \ j\%2==1$
 - $i\%2==1\ \&\& \ j\%2==1$
 - $i\%2==0\ \&\& \ j\%2==0$
 - $i\%2==1\ \&\& \ j\%2==0$
- Pentru a verifica dacă în tabloul unidimensional (2004,2008,2012,2016,2020,2024,2028,2032,2036,2040) există elementul $x=2023$ se aplică metoda căutării binare. Indicați numărul elementelor din tablou cu care se va compara x pe parcursul aplicării metodei.
 - 1
 - 2
 - 3
 - 4
- Variabila x memorează un număr întreg, indicați valoarea maximă pe care o poate avea expresia $(\text{abs}(x) - x) \% 100$.
 - 99
 - 98
 - 9
 - 0
- În secvența C/C++ alăturată toate variabilele sunt numere întregi. Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila nr să memoreze valoarea 246.

```
n=1234567, nr=0, p=1;
while(n>0){
    if(n%2==0){...}
    n=n/10;
}
```

 - $nr=p*(n\%10)+nr; p=p*10;$
 - $p=p*10; nr=p*(n\%10)+nr;$
 - $nr=(n\%10)*10+nr; p=p+1;$
 - $nr=(n\%10)*10+nr; p=p*10;$

SUBIECTUL al II-lea

(40 de puncte)

1. Algoritmul alăturat este reprezentat în pseudocod.
- a. Scrieți ce se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 6, 11, 5, 8, 13, 4, 2. (6p.)
- b. Scrieți un set de date de intrare, astfel încât, în urma executării algoritmului să se afișeze valoarea 0. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm echivalent cu cel dat, înlocuind structura **pentru...execută** cu o structură repetitivă cu test inițial. (6p.)
2. Pentru o melodie se memorează date specifice: **artist** (inițiala numelui interpretului – literă mare a alfabetului englez) și **an** (anul lansării melodiei – număr natural). Variabilele **artist1** și **an1** memorează date specifice pentru o melodie, iar variabilele **artist2** și **an2** memorează date specifice pentru altă melodie. Declarați variabilele **artist1**, **an1**, **artist2**, **an2** și scrieți o secvență de instrucțiuni, în urma executării căreia, să se afișeze pe ecran inițiala interpretului melodiei lansate mai repede sau mesajul egalitate dacă cele două melodii au fost lansate în același an. (6p.)
3. În secvența alăturată, **i** este de tip întreg iar variabila **c** este de tip **char**. Scrieți succesiunea de litere care se afișează în urma executării secvenței. (6p.)

```
citește n (număr natural nenul)
nr←0
pentru i←1,n execută
| citește z (număr natural nenul)
| x←1; y←1
| cât timp z>y execută
| aux←x; x←y; y←aux+y
| ■
| dacă z=aux+x atunci nr←nr+1
| ■
| ■
scrie nr
```

```
i=0;
while (i<6) {
    if (i%2==0 || i%3==0) c='a';
    else c='b';
    cout<<c; | printf("%c",c);
    i++;
}
```

SUBIECTUL al III-lea

(30 de puncte)

1. Un număr natural de 2 cifre se numește "dur" dacă este format doar cu cifre din intervalul $[1, 8]$, iar produsul cifrelor sale este mai mare sau egal cu 18.
Se citește un număr natural n ($n \in [1, 100)$) și se cere să se scrie, separate prin câte un spațiu, în ordine crescătoare, primele n numere dure. În cazul în care nu există suficiente numere "dure", se scrie mesajul "Nu exista".
Exemplu: dacă $n=10$ atunci, se scriu numerele 36 37 38 45 46 47 48 54 55 56
Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. (10p.)
2. Pe o hartă a unui joc pe calculator sunt dispuse mai multe încăperi alăturate care comunică prin uși montate pe fiecare perete vecin. Fiecare cameră conține un număr de comori care trebuie culese. Jucătorul este teleportat la început într-o cameră. El poate trece dintr-o cameră doar în camerele care comunică cu aceasta și are la dispoziție 2 încercări (stânga, dreapta). Jucătorul vrea să știe care este camera care i-ar aduce cel mai mare câștig (colectarea celor mai multe comori, inclusiv din camera în care se situează la un moment dat). Scrieți un program C/C++ care citește de la tastatură un număr natural n din intervalul $[2, 10^2]$ și apoi elementele unui tablou unidimensional, numere din intervalul $[0, 10]$, indexat de la 1, reprezentând numărul de comori din fiecare cameră. Programul afișează pe ecran poziția inițială de pe hartă, pentru încăperea care i-ar aduce jucătorului, numărul cel mai mare de comori. Dacă există mai multe soluții se va afișa prima dintre ele.
Exemplu: pentru $n=6$ și tabloul alăturat se va afișa indicele încăperii 2 (10p.)
- | | | | | | |
|---|---|---|---|---|---|
| 4 | 1 | 9 | 1 | 4 | 2 |
|---|---|---|---|---|---|
3. Se numește șir magic, un șir de numere consecutive în care fiecare valoare apare de exact două ori. Exemplu: șirul 2, 2, 3, 4, 3, 4 este un șir magic deoarece: numerele din șir fac parte dintr-o secvență de numere consecutive și fiecare dintre acestea apar în șir de exact două ori. Fișierul `magice.in` conține cel mult 10^9 numere naturale din intervalul $[1, 10^3)$. Numerele sunt separate prin câte un spațiu. Se cere să se afișeze pe ecran mesajul "Da" dacă numerele din fișier respectă proprietatea unui șir magic sau mesajul "Nu" în caz contrar. Proiectați un algoritm eficient din punctul de vedere al timpului de executare.
Exemplu: dacă fișierul `magice.in` conține numerele:
5 5 2 3 1 4 3 4 1 2
Se afișează pe ecran mesajul Da
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)