

Examenul de bacalaureat 2012
Proba E. d)
Proba scrisă la INFORMATICĂ
Limbajul C/C++

Varianta 2

Filiera teoretică, profilul real, specializările: matematică-informatică
matematică-informatică intensiv informatică
Filiera vocațională, profilul militar, specializarea matematică-informatică

- Toate subiectele sunt obligatorii. Se acordă 10 puncte din oficiu.
- Timpul efectiv de lucru este de 3 ore.
- În rezolvările cerute, identificatorii utilizați 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).
- În programele cerute, datele de intrare se consideră corecte, validarea acestora nefiind necesară.

SUBIECTUL I

(30 de puncte)

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

1. Variabilele x și y sunt de tip întreg ($0 < x < y$). Indicați expresia care are valoarea 1 dacă și numai dacă în intervalul deschis (x, y) **NU** sunt numere naturale pare. **(4p.)**

- a. $x \% 2 == 0 \ \&\& \ y - x <= 2 \ || \ x \% 2 == 1 \ \&\& \ y - x == 1$
- b. $x \% 2 == 0 \ || \ y - x <= 2 \ \&\& \ x \% 2 == 1 \ || \ y - x == 1$
- c. $x \% 2 == 0 \ \&\& \ y - x <= 2 \ \&\& \ x \% 2 == 1 \ \&\& \ y - x == 1$
- d. $x \% 2 == 0 \ || \ y - x <= 2 \ || \ x \% 2 == 1 \ || \ y - x == 1$

2. Se consideră algoritmul alăturat, reprezentat 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 valoarea afișată dacă se citesc, în această ordine, numerele 5, 6782, 123, 5421, 2, 78664. **(6p.)**

- b) Scrieți un șir de date de intrare, format doar din numere naturale cu cel mult două cifre fiecare, astfel încât în urma executării algoritmului, să se afișeze valoarea 2012. **(4p.)**

```
citește n (număr natural,  $n \leq 9$ )
m ← 0
cât timp n > 0 execută
| n ← n - 1
| citește x (număr natural)
| c ← 0
| cât timp  $x \% 10 < [x / 10] \% 10$  execută
| | c ←  $[x / 10] \% 10$ 
| | x ←  $[x / 10]$ 
| m ← m * 10 + c
scrie m
```

- c) Scrieți în pseudocod un algoritm, echivalent cu cel dat, în care să se înlocuiască prima structură **cât timp...execută** cu o structură repetitivă de tip **pentru ... execută**. **(6p.)**
- d) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

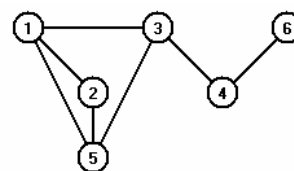
(30 de puncte)

Pentru fiecare dintre itemii 1 și 2 scrieți pe foaia de examen litera corespunzătoare răspunsului corect.

1. Se consideră arborele cu 7 noduri, numerotate de la 1 la 7, reprezentat prin vectorul de „tati” $(0, 1, 1, 3, 4, 4, 5)$. Nodurile de tip frunză ale arborelui sunt: **(4p.)**

- a. 1 b. 1,4,7
c. 2,6,7 d. 1,3,4,5

2. Se consideră graful neorientat cu șase noduri, reprezentat alăturat.
- Numărul minim de muchii ce pot fi eliminate, astfel încât în graful parțial obținut gradele nodurilor să fie egale, este: **(4p.)**



- a. 1 b. 2 c. 3 d. 4

Scrieti pe foaia de examen răspunsul pentru fiecare dintre cerințele următoare.

3. Se consideră declarările alăturate, în care variabilele **s1** și **s2** memorează coordonatele, în sistemul de coordonate **xOy**, ale extremităților câte unui segment. Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă segmentele corespunzătoare variabilelor **s1** și **s2** au comună extremitatea memorată în câmpul **A**. **(6p.)**

```
struct punct {
    int x,y;
};
struct segment {
    punct A,B;
} s1, s2;
```

4. În secvența de instrucțiuni de mai jos variabilele **i** și **j** sunt de tip întreg, iar variabila **A** memorează un tablou bidimensional cu 6 linii și 6 coloane, numerotate de la 1 la 6, cu elemente numere întregi.

Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot 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=1;i<=6;i++)
    for(j=1;j<=6;j++)
        .....
```

(6p.)

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

5. Scrieți un program C/C++ care citește de la tastatură două șiruri, fiecare fiind format din cel mult 30 de caractere. După primul șir se tastează Enter. Primul șir este format numai din litere mici ale alfabetului englez, iar cel de-al doilea șir este format numai din litere mici ale alfabetului englez și caractere *. Programul verifică dacă, înlocuind în cel de-al doilea șir toate caracterele * cu litere mici, șirul obținut este un subșir al primului șir. În caz afirmativ, programul afișează pe ecran șirul obținut, altfel afișează mesajul **imposibil**. Dacă se pot obtine mai multe astfel de șiruri, se afișează doar unul dintre ele.

Exemplu: dacă se citesc, în ordine, șirurile:

albastrui

a**r*

atunci pe ecran se afișează **astru**

(10p.)

SUBIECTUL al III-lea

(30 de puncte)

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

1. Utilizând metoda backtracking se generează, în ordine lexicografică, toate șirurile obținute prin permutarea literelor din mulțimea {b, r, a, d}. Primele două șiruri obținute sunt, în această ordine, **abdr** și **abrd**. Al 5-lea șir generat este: **(4p.)**

a. arbd b. ardb c. bdar d. bdra

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

2. Se consideră subprogramul **f**, definit alăturat. Scrieți ce se afișează în urma apelului de mai jos. **(6p.)**
- | | |
|------------------|--|
| <pre>f(9);</pre> | <pre>void f(int x) { if(x>0) { if(x%2==0) cout<<' '; printf(""); f(x/2); } cout<<x+1; printf("%d",x+1); }</pre> |
|------------------|--|
3. Se consideră subprogramul **elimina**, cu doi parametri:
- **n**, prin care primește o valoare naturală $2 < n < 50$;
 - **v**, prin care primește un tablou unidimensional cu **n** elemente, numere întregi cu cel mult 4 cifre. Cel puțin două dintre elementele alăturate ale tabloului sunt nule.
- Subprogramul elimină din tablou un număr minim de elemente astfel încât să nu existe două elemente alăturate cu valoarea 0. Tabloul modificat, precum și valoarea actualizată a lui **n**, sunt furnizate tot prin parametrii **v**, respectiv **n**.
Scrieți definiția completă a subprogramului.
Exemplu: dacă **n=11** și **v=(0, 2, 0, 0, 0, 0, -7, 0, 0, 3, 5)**, atunci după apel **n=7**, iar **v=(0, 2, 0, -7, 0, 3, 5)**. **(10p.)**
4. Fișierul **bac.txt** conține cel puțin două și cel mult 1000000 numere întregi, separate prin câte un spațiu. Fiecare număr are cel mult 9 cifre.
Se cere să se afișeze pe ecran, separate printr-un spațiu, cele mai mici două valori **distincte** memorate în fișier. Dacă în fișier nu există două astfel de valori, pe ecran se afișează mesajul **Nu exista**.
Pentru determinarea numerelor cerute se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.
Exemplu: dacă fișierul **bac.txt** conține valorile 26 -9023 29 -9023 90 3 234 se afișează pe ecran, nu neapărat în această ordine, numerele -9023 3.
a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**
b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**