

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

Varianta 4

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. Variabila **x** este de tip întreg și memorează un număr natural. ((x%5+1)%5+1)%5
Expresia alăturată are valoarea 0 dacă și numai dacă expresia
x%5 are valoarea: **(4p.)**

a. 1

b. 2

c. 3

d. 4

2. Se consideră algoritmul alăturat, reprezentat în pseudocod.

S-a notat cu **x%y** restul împărțirii numărului natural **x** la numărul natural nenul **y**.

- a) Scrieți valoarea afișată dacă se citesc, în această ordine, numerele 7, 5. (6p.)**
- b) Scrieți două seturi de date de intrare, formate din numere naturale cu cel mult două cifre fiecare, astfel încât în urma executării algoritmului, pentru fiecare dintre acestea, să se afișeze valoarea 28. (4p.)**

```
citește n,p (numere naturale nenule)
s←0
pentru i←1,n execută
|   j←i
|   cât timp j%p≠0 execută
|   |   j←j-1
|   ■
|   s←s+j
|   ■
scrie s
```

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

SUBIECTUL al II-lea

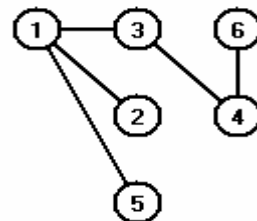
(30 de puncte)

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

1. Într-un arbore cu rădăcină considerăm că un nod se află pe nivelul x dacă lanțul elementar care are o extremitate în nodul respectiv și cealaltă extremitate în rădăcina arborelui are lungimea x . Pe nivelul 0 se află un singur nod (rădăcina).
Se consideră arborele cu 7 noduri, numerotate de la 1 la 7, reprezentat prin vectorul de „tați” $(0, 1, 1, 3, 4, 2, 3)$. Mulțimea tuturor nodurilor aflate pe nivelul 2 este: **(4p.)**

- a. $\{1, 2, 6\}$ b. $\{2, 3\}$ c. $\{4, 7\}$ d. $\{4, 6, 7\}$

2. Se consideră graful neorientat cu șase noduri, reprezentat alăturat.
Numărul minim de muchii ce trebuie adăugate, astfel încât în graful obținut fiecare nod să aparțină unui ciclu, este: **(4p.)**



- a. 1 b. 2 c. 4 d. 5

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

3. Se consideră declararea alăturată, în care variabilele $t1$ și $t2$ memorează lungimile laturilor a două triunghiuri.
Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă triunghiul corespunzător variabilei $t1$ are perimetrul strict mai mare decât perimetrul triunghiului corespunzător variabilei $t2$. **(6p.)**

```
struct triunghi {  
    int a,b,c;  
} t1, t2;
```

4. Se consideră două șiruri de caractere a , de lungime na și b , de lungime nb .
Șirul a este numit **sufix** al șirului b dacă $na \leq nb$ și subșirul lui b determinat de ultimele sale na caractere coincide cu a .
În secvența de instrucțiuni de mai jos variabila i este de tip întreg, iar variabila s memorează un șir cu cel mult 20 de caractere.
Fără a utiliza alte variabile, scrieți una sau mai multe instrucțiuni care pot înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran, în ordinea descrescătoare a lungimii, separate prin câte un spațiu, toate sufixele șirului memorat în variabila s , ca în exemplu.
Exemplu: pentru șirul `elevi` se afișează: `elevi levi evi vi i`
`for(i=0;i<strlen(s);i++)`
.....

(6p.)

5. Scrieți un program C/C++ care citește de la tastatură un număr natural n ($2 < n \leq 15$) și construiește în memorie un tablou bidimensional cu n linii și n coloane în care:
- toate elementele din prima coloană au valoarea 1;
- ultima linie conține, în ordine strict crescătoare, numerele naturale din intervalul $[1, n]$;
- oricare alt element este obținut prin însumarea celor două elemente vecine cu el, aflate pe linia imediat următoare și pe aceeași coloană cu el, respectiv pe aceeași linie cu el și pe coloana anterioară, ca în exemplu.

Programul afișează pe ecran tabloul obținut, fiecare linie a tabloului pe câte o linie a ecranului, elementele fiecărei linii fiind separate prin câte un spațiu.

Exemplu: pentru $n=4$ pe ecran se afișează tabloul alăturat. **(10p.)**

1	5	15	35
1	4	10	20
1	3	6	10
1	2	3	4

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 de maximum 4 litere distincte din mulțimea {a, v, i, o, n}. Primele 5 șiruri generate sunt, în această ordine: a, ai, ain, aino, ainv. Imediat după șirul avn se generează: **(4p.)**

a. avi

b. avni

c. avo

d. iano

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

2. Se consideră subprogramul `f`, definit alăturat. Scrieți valorile `f(3,4)` și `f(2,20)`. **(6p.)**
- ```
int f(int x,int y)
{ if(x*2>y)
 return x;
 return f(x*2,y);
}
```

3. Se consideră subprogramul `zero`, cu doi parametri:

- `n`, prin care primește o valoare naturală  $2 < n < 50$ ;
- `v`, prin care primește un tablou unidimensional cu  $2 \cdot n$  elemente, numere naturale cu cel mult 4 cifre. Numărul de elemente pare este egal cu numărul de elemente impare. Elementele au indici de la 1 la  $2 \cdot n$ .

Subprogramul modifică tabloul astfel încât elementele impare să aibă indici impari, iar elementele pare să aibă indici pari. Tabloul modificat este furnizat tot prin parametrul `v`.

Scrieți definiția completă a subprogramului.

**Exemplu:** dacă `n=5` și `v=(4, 5, 0, 9, 10, 7, 15, 3, 8, 10)`,

atunci după apel o soluție posibilă este: `v=(5, 4, 9, 0, 15, 10, 7, 8, 3, 10)`. **(10p.)**

4. Fișierul `bac.txt` conține pe prima linie un număr natural **par** `n` cu cel mult 3 cifre, iar pe următoarea linie un șir de `n` numere naturale cu cel mult nouă cifre. Numerele din șir sunt separate prin câte un spațiu.

Se cere să se afișeze pe ecran mesajul **Da** dacă în șir există un element care să fie strict mai mare decât jumătate dintre numerele din șir.

Dacă în șir nu se află o astfel de valoare, pe ecran se afișează mesajul **Nu**.

Pentru determinarea numărului cerut se utilizează un algoritm eficient din punctul de vedere al memoriei și al timpului de executare.

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

6

8 34 34 34 5 34

atunci pe ecran se afișează **Nu**, iar dacă fișierul are conținutul

8

1 5 6 12 3 12 12 9

atunci pe ecran se afișează **Da**.

a) Descrieți în limbaj natural algoritmul utilizat, justificând eficiența acestuia. **(4p.)**

b) Scrieți programul C/C++ corespunzător algoritmului descris. **(6p.)**