

# EXAMENUL DE BACALAUREAT

## Proba scrisă la INFORMATICĂ

### PROBA E, limbajul C/C++

#### Specializarea Matematică-informatică

**MODEL**

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, iar în lipsa unor precizări explicite, notațiile trebuie să corespundă cu semnificațiile asociate acestora

#### SUBIECTUL I (20 de puncte)

1. Indicați expresia C/C++ care are valoarea 1 dacă și numai dacă numărul natural memorat în variabila întregă  $n$  este divizibil cu 2, dar NU și cu 5. (4p.)

- a.  $!((n\%2==1) \ || \ (n\%5==0))$       b.  $(n\%2==0) \ \&\& \ (n\%5==0)$   
c.  $(n\%10==0) \ || \ (n\%5!=0)$       d.  $(n\%10==0) \ \&\& \ (n\%2==0)$

2. Variabila  $c$ , declarată alăturat, memorează titlul și prețul unei cărți. Expresia C/C++ a cărei valoare reprezintă prețul cărții respective majorat cu 50% este: (4p.)

```
struct carte { char titlu[21]; float pret; };
```

- a.  $c.pret*3/2$       b.  $pret.c*3/2$       c.  $c(pret)*3/2$       d.  $pret[c]*3/2$

3. Un arbore cu 37 de noduri, numerotate de la 1 la 37, are ca rădăcină nodul numerotat cu 1, iar tatăl fiecărui nod  $i$  ( $i \in [2,37]$ ) este numerotat cu partea întregă a rădăcinii pătrate a lui  $i$  ( $[ \sqrt{i} ]$ ). Numărul de frunze ale arborelui este: (4p.)

- a. 36    b. 31    c. 21    d. 6

4. Utilizând metoda backtracking, se generează toate parfumurile formate prin amestecarea a câte 3 esențe distincte din mulțimea {ambră, cedru, iris, mosc, santal}. Primele patru soluții obținute sunt, în această ordine: (ambră, cedru, iris), (ambră, cedru, mosc), (ambră, cedru, santal) și (ambră, iris, mosc). Indicați submulțimea care trebuie eliminată din enumerarea următoare, astfel încât soluțiile rămase să apară în ordinea generării lor: (ambră, mosc, santal), (cedru, mosc, santal), (cedru, iris, mosc), (cedru, iris, santal). (4p.)

- a. (ambră, mosc, santal)    b. (cedru, mosc, santal)    c. (cedru, iris, mosc)    d. (cedru, iris, santal)

5. Subprogramul  $F$  este incomplet definit alăturat. Indicați expresia cu care pot fi înlocuite punctele de suspensie, astfel încât, în urma apelului de mai jos, să se afișeze **dcba**.

```
F('a');      (4p.)
```

```
void F(char c)
{ if(c<'e')
  { F(.....);
  cout<<c; }
}
```

- a.  $c-4$     b.  $c-1$     c.  $c+1$     d.  $c+4$

## SUBIECTUL II (40 de puncte)

1. Se consideră algoritmul alăturat, reprezentat în pseudocod.

```
citește n,k
(numere naturale nenule)
cât timp n≥1 execută
┌─dacă n>k atunci ik
│alfel in
│■
n←n-i
t←k
┌─cât timp i≥1 execută
│scrie t,' '
│i←i-1
│t←t-1
│■
└─■
```

- Scrieți valorile afișate dacă se citesc, în această ordine, numerele 7 și 3. **(6p.)**
- Dacă pentru  $k$  se citește numărul **11**, scrieți cel mai mic și cel mai mare număr din intervalul **[1,99]** care pot fi citite pentru  $n$  astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, ultima valoare afișată să fie **7**. **(6p.)**
- Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind a doua structură cât timp...execută cu o structură repetitivă dealt tip. **(6p.)**
- Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**

2. Se consideră tipul elev, definit alăturat, ce permite declararea unor variabile care să memoreze anumite date despre un elev: numele și media sa generală anuală.

```
typedef struct { char nume[20];  
float medie; } elev;  
int i;
```

Utilizând tipul elev, declarați o variabilă, cu numele clasa, care să poată memora datele menționate mai sus pentru fiecare dintre cei 30 de elevi ai unei clase, apoi scrieți instrucțiunile care pot completa zona punctată din secvența de instrucțiuni de mai jos, pentru a inițializa, prin citire de la tastatură, variabila respectivă. **(6p.)**

```
for(i=0;i<30;i++) .....
```

3. Variabilele  $i$  și  $j$  sunt de tip întreg, iar variabila  $a$  memorează un tablou bidimensional cu 5 linii și 5 coloane, numerotate de la 1 la 5, având inițial toate elementele nule. Fără a utiliza alte variabile decât cele menționate, scrieți secvența de instrucțiuni de mai jos, înlocuind punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila  $a$  să memoreze tabloul alăturat. **(6p.)**

```
for(i=1;i<=5;i++)  
for(j=1;j<=5;j++)
```

```
1 2 3 4 5  
6 7 8 9 10  
11 12 13 14 15  
16 17 18 19 20  
21 22 23 24 25
```

### SUBIECTUL III (30 de puncte)

1. Sa se scrie un subprogram care primeste ca parametru un număr natural  $x$  ( $x > 1$ ) și care returneaza valoarea obtinuta prin însumarea exponentilor factorilor ce intră în descompunerea lui  $x$  în factori primi. (10p.)
2. Scrieti un program C/C++ care citește de la tastatură un număr natural nenul,  $n$  ( $n \leq 10$ ), și  $n$  cuvinte, fiecare cuvânt fiind format din cel puțin 3 și cel mult 20 de litere ale alfabetului englez. La introducerea datelor, după fiecare cuvânt se tastează Enter. Programul construiește în memorie, și apoi afișează pe ecran, șirul de caractere obtinut prin preluarea primelor trei litere ale fiecărui cuvânt citit, în ordinea citirii acestora. (10p.)  
Exemplu: pentru  $n=3$  și cuvintele frumos BUN Intelept se construiește șirul de caractere fruBUNInt
3. Fișierul bac.txt conține un șir de cel mult un milion de numere naturale din intervalul  $[0,102]$ , separate prin câte un spațiu. Se cere să se determine toate perechile distincte formate din termeni ai șirului aflat în fișier,  $x$  și  $y$  ( $y-x \geq 2$ ), astfel încât să nu existe niciun termen al șirului care să aparțină intervalului  $(x,y)$ . Numerele din fiecare pereche sunt afișate pe câte o linie a ecranului, în ordine strict crescătoare, separate printr-un spațiu, iar dacă nu există nicio astfel de pereche, se afișează pe ecran mesajul nu exista. Pentru determinarea numerelor cerute utilizați un algoritm eficient din punctul de vedere al timpului de executare.

Exemplu: dacă fișierul conține numerele

5 9 0 8 10 11 12 13 15 14 6 7 40 10 0 0 5 41 95 7 atunci pe ecran se afișează, nu neapărat în această ordine, perechile

0 5                    15 40                    41 95

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