

Examenul de bacalaureat național 2019  
Proba E. d)  
Informatică  
Limbajul C/C++

Varianta 1

Filieră teoretică, profil real, specializare științe ale naturii

- Toate subiectele sunt obligatorii. Se acordă 10 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.

**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  $x$  și  $y$  sunt întregi. Indicați expresia C/C++ echivalentă cu cea alăturată.

- a.  $!(x < 3) || (y >= 5)$                       b.  $!(x >= 3) \&\& (y < 5)$   
c.  $!(!(x < 3) || !(y >= 5))$                       d.  $!(x >= 3) \&\& (y < 5)$

2. Tabloul unidimensional  $v$  are elementele:

$v_1=4, v_2=3, v_3=2, v_4=1$ .

Știind că s-a notat cu  $x \leftrightarrow y$  interschimbarea valorilor variabilelor  $x$  și  $y$ , indicați cu ce se pot înlocui punctele de suspensie în algoritmul de sortare scris alăturat, pentru a ordona crescător cele patru elemente ale tabloului  $v$ .

```
pentru i ← 1, 3 execută  
  m ← i  
  pentru k ← i+1, 4 execută  
    dacă  $v_k < v_m$  atunci  
      m ← k  
    ■  
  ■  
  .....  
  ■
```

- a.  $v_i \leftrightarrow v_m$                       b.  $v_k \leftrightarrow v_m$                       c.  $v_k \leftrightarrow m$                       d.  $v_i \leftrightarrow m$

3. Variabilele  $i$  și  $j$  sunt de tip întreg. Indicați expresia care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, să se afișeze numerele de mai jos.

```
for(i=1; i<=5; i++)  
{ for(j=1; j<=5; j++)  
  if(.....) cout<<5-i<<' '; | printf("%d ", 5-i);  
  else cout<<5-j<<' '; | printf("%d ", 5-j);  
  cout<<endl; | printf("\n");  
}
```

```
4 4 4 4 4  
4 3 3 3 3  
4 3 2 2 2  
4 3 2 1 1  
4 3 2 1 0
```

- a.  $i < j$                       b.  $i > j$                       c.  $i + j < 5$                       d.  $i + j > 5$

4. Variabilele  $x$  și  $y$  sunt de tip real. Indicați instrucțiunea care realizează o prelucrare echivalentă cu cea alăturată.

```
if(x < 0) y = -x;  
else y = x;
```

- a.  $y = \text{sqrt}(x);$                       b.  $y = \text{abs}(x);$                       c.  $y = \text{floor}(x);$                       d.  $y = \text{ceil}(x);$

5. În secvența de mai jos toate variabilele sunt de tip întreg și memorează câte un număr natural nenul.

Indicați expresia care poate înlocui punctele de suspensie, astfel încât, în urma executării secvenței obținute, variabila  $cm$  să memoreze cel mai mare divizor comun al numerelor memorate în variabilele  $x$  și  $y$ .

```
d=1;  
while(.....)  
{ if(x%d==0 && y%d==0)  
  cm=d;  
  d=d+1;  
}
```

- a.  $d * d \leq x \ \&\& \ d * d \leq y$                       b.  $d \leq x / 2 \ \&\& \ d \leq y / 2$   
c.  $d > x \ \&\& \ d < y$                       d.  $d \leq x \ \&\& \ d \leq y$

**SUBIECTUL al II-lea**

**(40 de puncte)**

**1. Algoritmul alăturat este reprezentat în pseudocod.**

S-a notat cu  $a \% b$  restul împărțirii numărului natural  $a$  la numărul natural nenul  $b$  și cu  $[c]$  partea întregă a numărului real  $c$ .

```
citește n,k  
(numere naturale, k număr prim)  
p ← 0; i ← 1  
┌cât timp i ≤ n execută  
| x ← i  
| ┌cât timp x % k = 0 execută  
| | x ← [x/k]; p ← p + 1  
| | ──┐  
| | i ← i + 1  
| ──┐  
└─┘  
scrie p
```

- a) Scrieți valoarea afișată dacă se citește, în această ordine, numerele 10 și 3. **(6p.)**
- b) Dacă pentru  $k$  se citește numărul 5, scrieți trei numere care pot fi citite pentru  $n$  astfel încât, în urma executării algoritmului, pentru fiecare dintre acestea, valoarea afișată să fie 10. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind prima structură `cât timp...execută` cu o structură de tip `pentru...execută`. **(6p.)**
2. Variabilele `e_luna` și `e_zi`, de tip întreg, memorează luna și respectiv ziua desfășurării unui eveniment din anul 2019, iar variabilele `d_luna` și `d_zi` memorează luna și respectiv ziua corespunzătoare unei date calendaristice din același an. Scrieți o expresie C/C++ care are valoarea 1 dacă și numai dacă data calendaristică precizată este anterioară datei desfășurării evenimentului. **(6p.)**
3. Pentru a verifica dacă în tabloul unidimensional  $(0, 2, 5, 7, 16, 21, 49)$  există elementul cu valoarea  $x=15$ , se aplică metoda căutării binare. Scrieți succesiunea de elemente din tablou ale căror valori se compară cu valoarea lui  $x$  pe parcursul aplicării metodei indicate. **(6p.)**

**SUBIECTUL al III-lea**

**(30 de puncte)**

1. Se citește un număr natural,  $n$  ( $n \geq 10$ ), cu cel puțin o cifră impară, și se cere să se scrie valoarea 1 dacă toate cifrele impare ale lui  $n$  sunt egale între ele sau valoarea 0 în caz contrar. Scrieți, în pseudocod, algoritmul de rezolvare pentru problema enunțată.  
**Exemplu:** dacă  $n=7727470$  sau  $n=7240$  atunci se scrie valoarea 1, iar dacă  $n=7921470$  atunci se scrie valoarea 0. **(10p.)**
2. Scrieți un program C/C++ care citește de la tastatură un număr natural,  $n$  ( $n \in [2, 20]$ ), apoi cele  $n$  elemente ale unui tablou unidimensional, numere naturale din intervalul  $[0, 10^2)$ , dintre care cel puțin unul are o singură cifră și cel puțin unul are două cifre. Programul afișează pe prima linie a ecranului toate elementele tabloului formate din câte o singură cifră, iar pe a doua linie a ecranului celelalte elemente ale tabloului. Numerele de pe aceeași linie a ecranului sunt separate prin câte un spațiu.  
**Exemplu:** pentru  $n=6$  și tabloul  $(3, 14, 2, 1, 7, 21)$  se afișează pe ecran, nu neapărat în această ordine:  
3 2 1 7  
14 21 **(10p.)**
3. Se consideră șirul definit astfel:  $f_0=0$ ;  $f_1=x$ ;  $f_2=2 \cdot x - 1$ ;  $f_i=f_{i-1}+f_{i-2}-f_{i-3}$ , unde  $x$  și  $i$  sunt numere naturale nenule,  $i > 2$ . De exemplu, dacă  $x=5$  șirul este: 0, 5, 9, 14, 18, 23, 27, 32, 36, 41, ...  
Se citește de la tastatură două numere naturale,  $n$  și  $x$ , ( $n \in [1, 10^4]$ ,  $x \in [1, 10^2)$ ), unde  $x$  reprezintă al doilea termen al șirului precizat mai sus. Se cere să se scrie în fișierul `bac.txt` primii  $n$  termeni ai șirului, separați prin câte un spațiu, în ordine inversă a apariției lor în șir. Proiectați un algoritm eficient din punctul de vedere al timpului de executare și al spațiului de memorie utilizat.  
**Exemplu:** dacă  $n=8$  și  $x=5$ , atunci fișierul conține numerele 32 27 23 18 14 9 5 0  
a) Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. **(2p.)**  
b) Scrieți programul C/C++ corespunzător algoritmului proiectat. **(8p.)**