

**Examenul de bacalaureat național 2020**  
**Proba E. d)**  
**Informatică**  
**Limbajul C/C++**

Testul 13

*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.

- Indicați o expresie C/C++ care are valoarea 1 dacă și numai dacă numărul natural memorat în variabila întregă  $n$  este divizibil cu 4 și cu 5.
  - $!(n\%4!=0 \ || \ n\%5!=0)$
  - $!(n/4==1 \ || \ n/5!=0)$
  - $n\%4==0 \ \&\& \ !(n\%5==0)$
  - $n/4==0 \ \&\& \ !(n/5==0)$
- Pentru a verifica dacă într-un tablou unidimensional există elementul cu valoarea  $x=4$ , se aplică metoda căutării binare, iar succesiunea de elemente a căror valoare se compară cu  $x$  pe parcursul aplicării metodei este 14, 8, 4. Elementele tabloului pot fi:
  - (48, 14, 9, 8, 7, 4, 2)
  - (14, 14, 8, 8, 4, 4)
  - (14, 8, 4, 3, 2, 0)
  - (4, 8, 9, 14, 16, 24, 48)
- Tablourile unidimensionale  $A$  și  $B$  au elementele:  $A=(50, 36, 27, 20, 2)$ , iar  $B=(63, 45, 8, 5, 3)$ . În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:
  - (2, 3, 5, 8, 20, 27, 36, 45, 50, 63)
  - (2, 5, 8, 27, 50)
  - (50, 36, 27, 20, 2, 63, 45, 8, 5, 3)
  - (50, 63, 36, 45, 8, 27, 5, 20, 2, 3)
- O expresie C/C++ care are valoarea 1.0 este:
  - `ceil(-0.2080)`
  - `ceil(0.8020)`
  - `floor(-0.2080)`
  - `floor(0.8020)`
- În secvența de instrucțiuni alăturată, toate variabilele sunt de tip întreg. O expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței, variabila  $x$  să aibă o valoare egală cu cel mai mare divizor comun al numerelor 2020 și 2200 este:

	<code>x=2020; y=2200;</code> <code>while (y!=0)</code> <code>{ z=...; x=y; y=z; }</code>
<ol style="list-style-type: none"><li><math>x\%2</math></li><li><math>x/2</math></li><li><math>x\%y</math></li><li><math>x/y</math></li></ol>	

**SUBIECTUL al II-lea** (40 de puncte)

- 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 întreagă a numărului real  $c$ .
  - Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele 12345, 25, 2070, 49, 270135, 21. (6p.)
  - Dacă pentru  $n$  se citește numărul 100, scrieți un set de date din intervalul  $[10^3, 10^4)$  care pot fi citite în continuare, astfel încât, în urma executării algoritmului, să se afișeze 10. (6p.)
  - Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
  - Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să conțină o singură instrucțiune repetitivă. (6p.)

	<code>citește n (număr natural)</code> <code>p←1; m←0; k←0</code> <code>┌cât timp n≠0 execută</code> <code>│ citește x (număr natural)</code> <code>│ i←k</code> <code>└cât timp i≠0 execută</code> <code>│ x←[x/10]; i←i-1</code> <code>└─</code> <code>┌dacă x=0 atunci c←n%10</code> <code>│ altfel c←x%10</code> <code>└─</code> <code>m←c*p+m; n←[n/10]</code> <code>p←p*10; k←k+1</code> <code>└─</code> <code>scrie m</code>
--	---

2. Variabila întreagă **raza** memorează raza unui cerc, iar variabilele reale **centruX** și **centruY** memorează coordonatele reale (abscisa și ordonata), în planul **xOy**, ale centrului unui cerc. Declarați corespunzător variabilele **centruX** și **centruY** și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **Unitar** dacă cercul precizat are centrul în originea sistemului de coordonate și raza egală cu 1, sau mesajul **Altceva** în caz contrar. (6p.)
3. Variabila **i** este de tip întreg, iar celelalte variabile sunt de tip **char**. Scrieți ce se afișează în urma executării secvenței alăturate. (6p.)
- ```
c1='s';c2='o';c3='u';  
cout<<c2<<c3<<endl; | printf("%c%c\n",c2,c3);  
for(i=0;i<4;i++)  
{ c=c1+(i-1)*(1-i%2)+3*(2*i/3-1)*(i%2);  
  if(i==0) c1=c;  
  cout<<c; | printf("%c",c);  
}
```

### SUBIECTUL al III-lea

(30 de puncte)

1. Se citește un număr natural **n** ( $n \geq 2$ ) și se cere să se scrie, în această ordine, numerele **d** și **p**, reprezentând divizorul prim, **d**, care apare la cea mai mică putere, **p**, în descompunerea în factori primi a lui **n**; dacă există mai mulți astfel de divizori se afișează cel mai mic dintre ei. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **Exemplu:** dacă  $n=10780$ , atunci se scriu numerele 5 1 ( $10780=2^2 \cdot 5 \cdot 7^2 \cdot 11$ ). (10p.)
2. Scrieți un program C/C++ care citește de la tastatură două numere naturale din intervalul  $[2, 20]$ , **n** și **k**, și construiește în memorie un tablou unidimensional cu  $n \cdot k$  elemente astfel: tabloul memorează un șir crescător de termeni cu proprietatea că primul termen este **n**, fiecare valoare apare în șir de exact **k** ori și oricare doi termeni alăturați au valori egale sau consecutive. Programul afișează pe ecran tabloul construit, cu valorile separate prin câte un spațiu. **Exemplu:** dacă  $n=3$  și  $k=4$ , se obține tabloul alăturat. (10p.)
- ```
3 3 3 3 4 4 4 4 5 5 5 5 6 6 6 6
```
3. Șirul de mai jos este definit astfel:  $f_1=1$ ,  $f_n=2 \cdot f_{n-1}$  (unde **n** este un număr natural  $n \geq 2$ ).  
1, 2, 4, 8, 16, 32, ...  
Se citește de la tastatură un număr natural **x** ( $x \leq 10^9$ ), valoare a unui termen al șirului dat, și se cere să se scrie în fișierul text **bac.txt**, în ordine descrescătoare, separați prin câte un spațiu, toți termenii șirului care sunt mai mici sau egali cu **x**. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare. **Exemplu:** dacă se citește numărul 16  
fișierul **bac.txt** conține numerele 16 8 4 2 1  
a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)  
b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)