

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

Testul 20

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 x are exact două cifre.
 - $x/100!=0 \ || \ x/10==0$
 - $x/100==0 \ \&\& \ x/10!=0$
 - $x\%100!=0 \ || \ x\%10==0$
 - $x\%100==0 \ \&\& \ x\%10!=0$
- Pentru a verifica dacă într-un tablou unidimensional există elementul cu valoarea $x=3$, 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:
 - (14, 14, 8, 8, 4, 4)
 - (48, 14, 9, 8, 7, 4, 2)
 - (48, 24, 16, 14, 8, 4)
 - (48, 24, 16, 14, 9, 8, 4)
- Variabilele i și j sunt de tip întreg. Indicați expresia care poate înlocui zona punctată astfel încât, în urma executării secvenței obținute, să se afișeze pe ecran caracterele alăturate.

```
for(i=0;i<7;i++)  
{ for(j=0;j<7;j++)  
    if(.....) cout<<"a "; | printf("a ");  
    else cout<<"b "; | printf("b ");  
    cout<<endl; | printf("\n");  
}
```

b	a	a	a	a	a	b
b	b	a	a	a	b	b
b	b	b	a	b	b	b
b	b	b	b	b	b	b
b	b	b	a	b	b	b
b	b	a	a	a	b	b
b	a	a	a	a	a	b

 - $(i < j \ || \ i + j < 6) \ \&\& \ (i > j \ || \ i + j > 6)$
 - $(i < j \ \&\& \ i + j < 6) \ || \ (i > j \ \&\& \ i + j > 6)$
 - $(i < j \ || \ i + j < 6) \ || \ (i > j \ || \ i + j > 6)$
 - $(i < j \ \&\& \ i + j < 6) \ \&\& \ (i > j \ \&\& \ i + j > 6)$
- O expresie C/C++ care are valoarea 400 este:
 - `sqrt(20)`
 - `sqrt(40)`
 - `pow(20,2)`
 - `pow(10,4)`
- În secvența de instrucțiuni alăturată, toate variabilele sunt întregi. Indicați o expresie care poate înlocui punctele de suspensie astfel încât, în urma executării secvenței obținute, variabila x să aibă o valoare egală cu cel mai mare divizor comun al numerelor 50 și 2020.

```
x=2020; y=50;  
while (x!=y)  
{ if(x>y) x=x-y;  
  else y=.....;  
}
```

 - $y-x$
 - $2*x-y$
 - $y-x/2$
 - $(x-y)/2$

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 întreagă a numărului real c .
- a. Scrieți ce se afișează dacă se citește numărul 12. (6p.)
- b. Scrieți două numere din intervalul $[2, 10^2]$, unul par și unul impar, care pot fi citite astfel încât, pentru fiecare dintre acestea, în urma executării algoritmului, să se afișeze 9. (6p.)
- c. Scrieți programul C/C++ corespunzător algoritmului dat. (10p.)
- d. Scrieți în pseudocod un algoritm, echivalent cu cel dat, înlocuind adecvat structura `cât timp...execută` cu o structură repetitivă de alt tip. (6p.)
2. Pentru fiecare ingredient utilizat pentru prepararea unei prăjituri se memorează un cod și cantitatea necesară. Codul reprezintă o literă, corespunzătoare numelui ingredientului: **C** pentru cacao, **M** pentru miere etc. Variabilele `cod1` și `cantitate1` memorează codul și cantitatea corespunzătoare unui ingredient, iar variabilele `cod2` și `cantitate2` memorează codul și cantitatea corespunzătoare unui alt ingredient. Știind că ingredientele sunt în cantități diferite, scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran codul ingredientului din care este necesară o cantitate mai mare, urmat, pe linia următoare, de numele acestuia, dacă este miere sau cacao, sau de mesajul `altceva` dacă este un alt ingredient. (6p.)
3. Știind că unul dintre cele trei elemente ale tabloului **A** are valoarea 45 și două dintre elementele tabloului **B** au valorile 21, respectiv 52, scrieți câte un exemplu de valori pentru elementele tabloului **A**, respectiv ale tabloului **B**, în ordinea în care ele pot apărea în fiecare tablou, astfel încât, prin metoda interclasării descrescătoare a acestora, să se obțină tabloul unidimensional (52, 45, 44, 21, 10, 5, 2). (6p.)

```
citește n
(număr natural nenul)
d←1; s←0
cât timp d*d<n execută
┌dacă n%d=0 și d%2≠[n/d]%2 atunci
│ s←s+d+[n/d]
│ ──┘
│ d←d+1
│ ──┘
└dacă d*d=n atunci s←s+d
  ──┘
scrie s
```

SUBIECTUL al III-lea

(30 de puncte)

1. Se citesc două numere naturale, b și n ($b \in [2, 10]$, $n \in [0, 10^9]$) și se cere să se afișeze suma tuturor produselor de forma $c \cdot b^k$, unde c este cifra de pe poziția k în scrierea numărului n ; pozițiile sunt numerotate de la dreapta la stânga, iar cifra unităților este pe poziția 0. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **Exemplu:** dacă $b=2$ și $n=10010$, se scrie numărul 18 ($18=1 \cdot 2^4+0 \cdot 2^3+0 \cdot 2^2+1 \cdot 2^1+0 \cdot 2^0$). (10p.)
2. Scrieți un program C/C++ care citește de la tastatură un număr natural, n ($n \in [2, 20]$), și cele n elemente ale unui tablou unidimensional, numere reale pozitive. Programul transformă tabloul în memorie, înlocuind fiecare număr real cu partea întreagă a acestuia. **Exemplu:** pentru $n=10$ și tabloul (10, 20.64, 25, 40.45, 50, 80.21, 40.56, 1.2, 10.6, 5) se obține tabloul (10, 20, 25, 40, 50, 80, 40, 1, 10, 5) (10p.)
3. Fișierul `bac.txt` conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^3]$, separate prin câte un spațiu. Se cere să se afișeze pe ecran suma maximă obținută adunând numai numere pare de pe poziții consecutive în șirul aflat în fișier. Proiectați un algoritm eficient din punctul de vedere al memoriei utilizate și al timpului de executare. **Exemplu:** dacă fișierul `bac.txt` conține valorile 10 115 1 5 2 2 2 4 7 3 100 20 2 7 se afișează pe ecran numărul 122
- a. Descrieți în limbaj natural algoritmul proiectat, justificând eficiența acestuia. (2p.)
- b. Scrieți programul C/C++ corespunzător algoritmului proiectat. (8p.)