

MODEL TEST 5

- 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 se notează cu 4 puncte.

1. Variabila **n** memorează un număr natural cu 5 cifre. Care dintre expresiile C/C++ de mai jos schimbă cifra din mijloc cu cifra 0?

- | | |
|--|---|
| a) $n = n \% 1000 + n / 100 * 100$ | c) $n = n \% 100 + n / 1000 * 1000$ |
| b) $n = n \% 10 + n / 10 * 10 + n / 100 * 100$ | d) $n = n / 10 \% 100 * 1000 + n / 100$ |

2. Pentru a verifica dacă în tabloul unidimensional (**6, 7, 9, 11, 12, 19, 28**), există elementul cu valoarea **x=18** se aplică metoda căutării binare. Succesiunea de elemente a căror valoare se compară cu x pe parcursul aplicării metodei este:

- | | |
|---------------|---------------|
| a) 11, 9, 6 | c) 11, 19, 28 |
| b) 11, 19, 12 | d) 11, 12, 19 |

3. Se consideră două tablouri unidimensionale **A = (1, 3, 5, 9, 10)** și **B = (2, 4, 6, 7)**. În urma interclasării lor în ordine crescătoare se obține tabloul cu elementele:

- | | |
|---------------------------------|--------------------------------------|
| a) (1, 2, 3, 4, 5, 6, 9, 7, 10) | c) Nu se poate realiza interclasarea |
| b) (1, 2, 3, 4, 5, 6, 7, 9, 10) | d) (1, 3, 5, 9, 10, 2, 4, 6, 7) |

4. Care dintre expresiile C/C++ de mai jos are valoarea 1?

- a) $2.17 \geq \text{floor}(2.17) \ \&\& \ 2.17 \leq \text{ceil}(2.17)$
 b) $2.17 \leq \text{floor}(2.17) \ \&\& \ 2.17 + 1 == \text{ceil}(2.17)$
 c) $2.17 < \text{floor}(2.17) \ \&\& \ (2.17) > 1 + \text{floor}(2.17)$
 d) $2.17 > \text{floor}(20.17) \ \&\& \ 2.17 - 1 == \text{floor}(20.17)$

5. Câte atribuiri se execută în total, în secvența alăturată, dacă **n** și **p** sunt variabile de tip întreg?

- | | |
|------|-------|
| a) 7 | c) 9 |
| b) 8 | d) 10 |

```

p=1; n=52;
while(n>=p)
p=p*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 b .

- a) Scrieți valoarea care se afișează în urma executării algoritmului dacă se citesc, în această ordine, numerele **4, 3, 12, 36, 45, 51, 27, 87, 17, 25**. **(6p.)**
- b) Dacă pentru n se citește 2, iar pentru p se citește 5, scrieți patru numere distincte care pot fi citite, astfel încât în urma executării algoritmului, valoarea afișată să fie **0**. **(6p.)**
- c) Scrieți programul C/C++ corespunzător algoritmului dat. **(10p.)**
- d) Scrieți în pseudocod un algoritm, echivalent cu cel dat, care să înlocuiască structura **repetă...până când** cu o structură de tip **pentru...execută**. **(6p.)**

```

citește n, p
(numere naturale nenule)
nr ← 0
repetă
    citește x, y
    (numere naturale)
    cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    ■ dacă (x=p) atunci
        nr ← nr + 1
    ■
n ← n - 1
până când n = 0
scrie nr

```

2. Variabilele reale **nota1**, **nota2** și **nota3** memorează notele obținute de un elev la cele trei probe ale examenului de bacalaureat. Declarați corespunzător variabilele și scrieți o secvență de instrucțiuni în urma executării căreia să se afișeze pe ecran mesajul **reusit** și media obținută, dacă elevul este promovat la examen, sau mesajul **respins** în caz contrar (un elev este promovat dacă are cel puțin nota 5 la fiecare probă și media cel puțin 6).

(6p.)

3. În secvența de instrucțiuni de mai jos variabilele i și j sunt de tip întreg. 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 numerele de mai jos, în această ordine.

(6p.)

for(i=1;i<=4;i++){	0	1	1	1	1	1
for (j=1;j<=6;j++)	1	0	1	0	1	0
.....}	0	1	0	3	3	3
	1	2	3	0	1	2

SUBIECTUL al III-lea**(30 de puncte)**

1. Un număr natural n , având un număr de k cifre, se numește număr **Armstrong** dacă este egal cu suma cifrelor sale ridicate la puterea k .

Se citește un număr natural n , și se cere să se scrie valoarea **1** dacă n este număr Armstrong sau valoarea **0** în caz contrar. Scrieți, în pseudocod, algoritmul de rezolvare a problemei enunțate. **(10p.)**

Exemplu: dacă $n=153$ se scrie valoarea 1 ($153=1^3+5^3+3^3$).

2. Scrieți un program C/C++, care citește de la tastatură un număr natural, n ($n \in [2, 10^2]$), apoi cele n elemente ale unui tablou unidimensional, numere naturale. Programul transformă în memorie tabloul, eliminând din componența sa toate numerele care sunt puteri ale lui 2, apoi afișează pe ecran elementele tabloului astfel obținut. Dacă nu există niciun astfel de număr, se va afișa mesajul **nu exista**. Dacă toate elementele tabloului sunt puteri ale lui 2, se va afișa mesajul **s-au eliminat toate elementele**. **(10p.)**

Exemplu: pentru $n=10$ și tabloul (23, **32**, 45, 56, **64**, 11, 25, **128**, 78, **8**) după execuția programului tabloul va fi (23, 45, 56, 11, 25, 78).

3. Numerele întregi pozitive cu proprietatea că, prin însumarea iterativă a pătratelor cifrelor lor, se ajunge în cele din urmă la numărul 1, se numesc **numere fericite**. Numărul 7 este un număr fericit pentru că $7^2=49$, $4^2 + 9^2=97$, $9^2 + 7^2=130$, $1^2 + 3^2 + 0^2=10$, $1^2 + 0^2=1$. Prelucrând astfel orice număr, în cele din urmă se va ajunge doar la unul dintre următoarele numere posibile: 0, 1, 4, 16, 20, 37, 42, 58, 89 sau 145.

Fișierul **bac.in** conține un șir de cel mult 10^6 numere naturale din intervalul $[0, 10^9]$, separate prin câte un spațiu. Se cere să se afișeze în fișierul **bac.out**, pe câte un rând, numerele din șir care sunt *fericite* urmate de numărul de iterații necesare pentru a ajunge la numărul 1. Dacă în șir nu există niciun număr fericit se va afișa mesajul **nu exista**. Proiectați un algoritm eficient din punct de vedere al timpului de executare și al memoriei utilizate.

Exemplu:

bac.in	bac.out
7 13 95 104 86 17 379 226 445 33	7 5
	13 2
	86 2
	379 6
	226 5

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

REZOLVARE SUBIECTE

TEST 5

Subiectul I

1. c
2. b
3. b
4. a
5. b

Subiectul II

1. Rezolvare:

a) 2

n	p	nr	x	y	z
4	3	0	12	36	12
			36	12	0
			12	0	
3		1	45	51	45
			51	45	6
			45	6	3
			6	3	0
			3	0	
2		2	27	87	27
			87	27	6
			27	6	3
			6	3	0
			3	0	
1			17	25	17
			25	17	8
			17	8	1
			8	1	0
			1	0	
0					

b) Algoritmul numără câte seturi de numere x, y au cel mai mare divizor comun numărul p .

Un set de numere posibile: 64, 26, 13, 29

c) Programul C++:

```
#include <iostream>
using namespace std;
int main(){
    int n,p,nr,x,y,z;
    cin>>n>>p;
    nr=0;
    do{
        cin>>x>>y;
        while(y!=0){
            z=x%y;
            x=y;
            y=z;
        }
        if(x==p)
            nr++;
        n--;
    }while(n!=0);
    cout<<nr;
    return 0;
}
```

d) Pseudocod echivalent:

```
citește n, p (numere naturale)
nr ← 0
pentru i ← 1, n, 1 execută
    citește x, y (numere naturale)
    cât timp y ≠ 0 execută
        z ← x % y
        x ← y
        y ← z
    dacă (x = p) atunci
        nr ← nr + 1
scrie nr
```

2.

```
float nota1, nota2, nota3, media;
if(nota1>=5 && nota2>=5 && nota3>=5){
    medie=(nota1+nota2+nota3)/3;
    if(medie>=6)
        cout<<"reusit "<<medie;
    else
        cout<<"respins";
}
else cout<<"respins";
```
3.

```
for(i=1;i<=4;i++){
    for(j=1;j<=6;j++){
        if(i%2==1) cout<<i%j<<' ';
        else cout<<j%i<<' ';
    }
    cout<<endl;}
```

Subiectul III

1. Rezolvare:

```

citește n (număr natural)
k ← 0
aux ← n
cât timp (aux ≠ 0) execută
    k ← k + 1
    aux ← [aux / 10]
aux ← n
s ← 0
cât timp (aux ≠ 0) execută
    p ← 1
    pentru i ← 1, k, 1 execută
        p ← p * (aux % 10)
    s ← s + p
    aux ← aux / 10
dacă (n = s) atunci
    scrie 1
altfel
    scrie 0

```

2. Rezolvare:

```

#include <iostream>
using namespace std;
int main(){
    int n, i, j, v[101], aux, ok=0;
    cin >> n;
    for(i=1; i<=n; i++){
        cin >> v[i];
    }
    for(i=1; i<=n; i++){
        aux = v[i];
        while(aux % 2 == 0){
            aux = aux / 2;
        }
        if(aux == 1){
            ok = 1;
            for(j=i; j<=n-1; j++){
                v[j] = v[j+1];
            }
            n--; i--;
        }
    }
    if(ok == 0) cout << "nu exista";
    else{
        if(n == 0)
            cout << "s-au eliminat toate elementele";
        else
            for(i=1; i<=n; i++)
                cout << v[i] << ' ';
    }
    return 0;
}

```

3. Rezolvare:

```
#include <iostream>
#include<fstream>
using namespace std;
int main(){
    ofstream fout("bac.out");
    ifstream fin ("bac.in");
    int s,x,it,aux,ok;
    ok=0;
    while(fin>>x){
        aux=x; it=0;
        do{
            s=0; it++;
            while(x!=0){
                s=s+x%10*(x%10);
                x=x/10;
            }
            x=s;
        }while(s!=1 && s!=0 && s!=4 && s!=16 && s!=20 && s!=37 && s!=42
&& s!=58 && s!=89 && s!=145);
        if(s==1){
            fout<<aux<<' '<<it<<endl;
            ok=1;
        }
    }
    if(ok==0)
        fout<<"nu exista";
    return 0;
}
```

BAREM DE EVALUARE ȘI DE NOTARE**TEST 5**

- Se punctează oricare alte modalități de rezolvare corectă a cerințelor.
- Nu se acordă punctaje intermediare, altele decât cele precizate explicit prin barem. Nu se acordă fracțiuni de punct. Se acordă 10 puncte din oficiu. Nota finală se calculează prin împărțirea punctajului total acordat pentru lucrare la 10.
- Utilizarea unui tip de date care depășește domeniul de valori precizat în enunț este acceptată dacă acest lucru nu afectează corectitudinea în funcționarea programului

SUBIECTUL I**(20 de puncte)**

1c 2b 3b 4a 5b	5 x 4p.
-----------------------	----------------

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

1.	a) Răspuns corect: 2	6p.	
	b) Pentru răspuns corect	6p.	Se acordă câte 3p. pentru fiecare set de numere x, y conform cerinței (un set de numere x, y este corect dacă c.m.m.d.c (x,y) este diferit de 5).
	c) Pentru program corect -declarare variabile -citire date -afișare date -instrucțiune de decizie -instrucțiuni repetitive (*) -atribuiri -corectitudine globală a programului ¹⁾	10p. 1p. 1p. 1p. 2p. 3p. 1p. 1p.	(*) Se acordă numai 2p. dacă doar una dintre instrucțiunile repetitive este conform cerinței.
	d) Pentru algoritm pseudocod corect -echivalență a prelucrării realizate, conform cerinței (*) - corectitudine globală a algoritmului ¹⁾	6p. 5p. 1p.	(*) Se acordă numai 2p. dacă algoritmul are o structură repetitivă conform cerinței, principial corectă, dar nu este echivalent cu cel dat. Se va puncta orice formă corectă de structură repetitivă conform cerinței.
2.	Pentru rezolvare corectă -definire a variabilelor conform cerinței -verificare a condiției impuse (*) -afișare mesaj -corectitudine globală a expresiei ¹⁾	6p. 1p. 3p. 1p. 1p.	(*) Se acordă câte 1p. pentru fiecare aspect al cerinței referitor la condiția impusă (note, medie, operatori logici utilizați conform cerinței).
3.	Pentru rezolvare corectă -atribuire a valorilor indicate pentru afișare (*) -afișarea valorilor pe linii, conform cerinței -corectitudine globală a secvenței ¹⁾	6p. 4p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect specific (afișare valori, identificare linii cu indice par/linii cu indice impar). O soluție posibilă este afișarea valorii expresiei $i\%j$ elementelor aflate pe linii cu indice impar, respectiv $j\%i$ elementelor aflate pe linii cu indice par.

SUBIECTUL al III-lea**(30 de puncte)**

1.	Pentru algoritm corect -citire a datelor -verificare a proprietății cerute (*) -scriere a datelor -scriere principial corectă a structurilor de control, corectitudine globală a algoritmului ¹⁾ (**) 	10p. 1p. 6p. 1p. 2p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (identificare a unei cifre, ridicarea unui număr la o putere, calcul sumă). (**) Se va puncta orice formă corectă de structură repetitivă sau decizională.
2.	Pentru program corect -declarare a unei variabile care să memoreze un tablou unidimensional -citire a datelor -transformare a tabloului (*) -afișare a datelor și tratare a cazurilor nu exista/s-au eliminat toate elementele -declarare a variabilelor simple, corectitudine globală a programului ¹⁾ 	10p. 1p. 1p. 6p. 1p. 1p.	(*) Se acordă câte 2p. pentru fiecare aspect al cerinței (eliminarea a unui element, algoritm principial corect de verificare a unui număr dacă este putere a lui 2, transformare în memorie).
3.	a) Pentru program corect -operații cu fișiere: declarare, pregătire în vederea citirii/scrierii, citire/scriere în fișier -determinare a valorilor cerute (*), (**) -utilizare a unui algoritm eficient (***) -declarare a variabilelor, citire a datelor, corectitudine globală a programului b) Pentru răspuns corect -coerență a descrierii algoritmului (*) -justificare a elementelor de eficiență	8p. 1p. 5p. 1p. 1p. 2p. 1p. 1p.	(*) Se acordă punctajul chiar dacă algoritmul ales nu este eficient. (**) Se acordă numai 3p. dacă algoritmul este principial corect, dar nu oferă rezultatul cerut pentru toate seturile de date de intrare. (***) Se acordă punctajul numai pentru un algoritm liniar care utilizează eficient memoria. O soluție posibilă parcurge șirul din fișier, memorând valoarea curentă, calculează iterativ suma pătratelor cifrelor cât timp nu s-a ajuns la o valoare particulară și numără iterațiile efectuate.

¹⁾ Corectitudinea globală vizează structura, sintaxa, alte aspecte neprecizate în barem.