

Admitere Mate-Info - model
Proba scrisă la Informatică

Subiectul A (30 puncte)

1. (5p) Un tip de date întreg pe x biți (x este număr natural strict pozitiv) va putea reține valori întregi din intervalul:

- a. $[0, 2^x]$
- b. $[0, 2^{x-1}-1]$
- c. $[-2^{x-1}, 2^{x-1}-1]$
- d. $[-2^x, 2^x-1]$
- e. $[0, 10^x]$

2. (5p) Se dă următorul subalgoritm

```
Subalgoritm f(a, b):  
    Dacă a > 1 atunci  
        returnează b * f(a - 1, b)  
    altfel  
        returnează b * f(a + 1, b)  
SfDacă  
SfSubalgoritm
```

Precizați de câte ori se apelează funcția f în următoarea secvență de cod:

- | | |
|------------------------|---------------------------|
| $x \leftarrow 4$ | a. de 4 ori |
| $y \leftarrow 3$ | b. de 3 ori |
| $z \leftarrow f(x, y)$ | c. de o infinitate de ori |
| | d. niciodată |
| | e. o dată |

3. (5p) Fie x o variabilă de tip întreg care conține cel mai mic număr natural nenul, multiplu de 36, divizibil cu toate numerele prime mai mici decât 10. Precizați care dintre expresiile de mai jos sunt adevărate.

- a. $(x < 1000) \text{ and } ((x*x*x) \bmod 1000 = 0)$
- b. $(x \bmod 100 = 0) \text{ or } (x \text{ div } 100 = 0)$
- c. $(x > 1000) \ \&\& \ (x \bmod 7 = 0)$
- d. $((x*x) \text{ div } 16) \bmod 2 = 1$
- e. $((x*x) \text{ div } 16) \bmod 2 = 0$

4. (5p) Se consideră toate șirurile de lungime $l \in \{1, 2, 3\}$ formate din litere din mulțimea $\{a, b, c, d, e\}$. Câte dintre aceste șiruri au elementele ordonate strict descrescător și un număr impar de vocale? (a și e sunt vocale)

- a. 14
- b. 7
- c. 81
- d. 78
- e. 0

5. (5p) Se dă următoarea secvență de cod

```
void numerePozitive(int m, int a[],  
int &n, int b[]){  
    n = 0;  
    for(int i = 0; i < n; i++){  
        procedure numerePozitive(m:integer; a:sir;  
var n:integer; var b:sir)  
            Begin  
                n := 0;
```

```

    if (a[i] > 0){
        n = n + 1;
        b[n] = a[i];
    }
}

for i := 1 to n do
    if (a[i] > 0) then
        begin
            n := n + 1;
            b[n] := a[i];
        end;
End;

```

Care este rezultatul execuției apelului $numerePozitive(k,x,l,y)$ pentru $k=4$, șirul $x=(-1,2,-3,4)$, $l = -1$ și șirul vid $y = ()$.

- $l = 3$ și $y=[2, 4]$;
- $l = 0$ și $y=[2, 4]$;
- $l = 0$ și $y=[]$;
- Depinde de valoarea lui k
- Eroare de compilare

6. (5p) Se consideră următorul subalgoritm:

```

Subalgoritm SA6(a):
    Dacă a < 50 atunci
        Dacă a mod 3 = 0 atunci
            returneaza SA6(2 * a - 3)
        altfel
            returnează SA6(2 * a - 1)
    SfDacă
    altfel
        returnează a
    SfDacă
SfSubalgoritm

```

Pentru care dintre valorile parametrului a subalgoritmul va returna valoarea 61?

- 16
- 61
- 4
- 31
- 51

Subiectul B (60 puncte)

1. Degustare de ciocolată (25 puncte)

O companie de publicitate face reclamă la un nou sortiment de ciocolată și intenționează să distribuie mostre de ciocolată la n ($10 \leq n \leq 10\,000\,000$) copii care sunt așezați într-un cerc. Angajații companiei își dau seama că distribuirea de mostre tuturor copiilor ar costa foarte mult. În consecință, decid să distribuie mostre fiecărui al k -lea ($0 < k < n$) copil din cei n , numărând copiii din k în k (atunci când numărătoarea ajunge la ultimul copil, ea continuă cu primul copil și așa mai departe). În numărătoare se vor considera toți copiii, fie că au primit sau nu ciocolată. Numărătoarea *se oprește atunci când o ciocolată ar trebui distribuită unui copil care deja a primit*.

Scrieți un subalgoritm care determină numărul copiilor (nr) care nu primesc mostre de ciocolată. Parametrii de intrare sunt numerele naturale n și k , iar parametrul de ieșire va fi numărul natural nr .

Exemplu 1: dacă $n = 12$ și $k = 9$, atunci $nr = 8$ (primul, al 2-lea, al 4-lea, al 5-lea, al 7-lea, al 8-lea, al 10-lea, al 11-lea copil nu primesc ciocolată).

Exemplu 2: dacă $n = 15$ și $k = 7$, atunci $nr = 0$ (toți copiii primesc ciocolată).

2. Numere magice (15 puncte)

Se consideră două numere naturale p și q ($2 \leq p \leq 10$, $2 \leq q \leq 10$). Un număr natural se numește *magic* dacă mulțimea cifrelor utilizate în scrierea lui în sistemul de numerație având baza p este identică cu mulțimea cifrelor folosite în scrierea lui în sistemul de numerație având baza q . De exemplu, pentru $p = 9$ și $q = 7$, $(31)_{10}$ este număr *magic* pentru că $(34)_9 = (43)_7$, iar pentru $p = 3$ și $q = 9$, $(9)_{10}$ este număr *magic* pentru că $(100)_3 = (10)_9$.

Scrieți un subalgoritm care, pentru două baze p și q date determină șirul x al tuturor numerelor *magice* strict mai mari ca 0 și strict mai mici decât un număr natural n dat ($1 < n \leq 10000$). Parametrii de intrare ai subalgoritmului sunt p și q (cele două baze) și valoarea n . Parametrii de ieșire vor fi șirul x și lungimea k a șirului x .

Exemplu: dacă $p = 9$, $q = 7$ și $n = 500$, șirul x va avea $k = 11$ elemente: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

3. Căutare (10 puncte)

Se dă următorul subalgoritm:

```
1: Subalgoritm cautare(x, n, val):
2:     Dacă n = 0 atunci
3:         returnează (x[0] = val)
4:     altfel
5:         returnează cautare(x, n - 1, val)
6:     SfDacă
7: SfSubalgoritm
```

Ce instrucțiune sau instrucțiuni trebuie adăugate și unde astfel încât în urma apelului, funcția să determine dacă elementul *val* face sau nu parte din șirul x cu n elemente (n număr natural strict mai mare ca zero)?

4. Cifra de control (10 puncte)

Se dă următorul subalgoritm pentru determinarea cifrei de control a unui număr natural cu minim 2 cifre.

```
1: Subalgoritm cifraDeControl(x):
2:     CâtTimp x > 9 execută:
3:         s ← 0
4:         CâtTimp x > 0 execută:
5:             s ← s + x MOD 10 { x mod 10 calculează restul împărțirii lui x la 10}
6:             x ← x DIV 10 { x div 10 calculează câtul împărțirii lui x la 10}
7:         SfCâtTimp
8:         x ← s
9:     SfCâtTimp
10:     returnează x
11: SfSubalgoritm
```

Înlocuiți corpul acestui subalgoritm cu maxim 2 instrucțiuni astfel încât subalgoritmul rezultat să aibă același efect.

Notă:

1. Toate subiectele sunt obligatorii.
2. Rezolvările trebuie scrise detaliat pe foile de examen (ciornele nu se iau în considerare).
3. Se acordă 10 puncte din oficiu.
4. Timpul efectiv de lucru este de 3 ore.

BAREM

OFICIU..... 10 puncte

SUBIECTUL A 30 puncte

A. 1. Răspunsul b,c..... 5 puncte

A. 2. Răspunsul c..... 5 puncte

A. 3. Răspunsurile c,d..... 5 puncte

A. 4. Răspunsul a..... 5 puncte

A. 5. Răspunsul c..... 5 puncte

A. 6. Răspunsurile a, b, d..... 5 puncte

SUBIECTUL B 60 puncte

B. 1. Degustare de ciocolată 25 puncte

- V1: determinarea corectă a valorii nr (cu formula $nr = n - n/cmmdc(n, k)$)..... 25 puncte
- V2: determinarea corectă a valorii nr (simulare, listă circulară) 15 puncte

B. 2. Numere magice..... 15 puncte

- verificarea proprietății de *număr magic*
 - V1: pe baza identității vectorilor caracteristici ai mulțimilor de cifre ale numărului dat în cele două reprezentări (în baza p și, respectiv, baza q) 10 puncte
 - V2: alte variante de algoritm corect cu performanță mai redusă maxim 5 puncte
- construirea șirului x 5 puncte

B. 3. Căutare 10 puncte

- identificare condiție ($x[n] = val$) 5 puncte
- returnarea valorii de adevăr a condiției compuse în linia 5 5 puncte

B. 4. Cifra de control 10 puncte

- cifra de control a unui număr poate fi calculată ca $nr \bmod 9$ 10 puncte

REZOLVARE – Subiect B.1.: Degustare de ciocolată.....25 puncte

```

#include <iostream>
using namespace std;
/*****
Subiectul I.1. Degustare de ciocolata
*****/
//calculeaza si returneaza cmmdc a 2 numere naturale a si b
int cmmdc(int a, int b){
    if ((a == b) && (a == 0))
        return 1;
    if (a * b == 0)
        return a + b;
    while (b != 0){
        int c = b;
        b = a % b;
        a = c;
    } //while
    return a;
}
//determina si returneaza nr de copii care nu primesc ciocolata dintre cei n copii
//numarand din k in k. Putem să considerăm număratoarea în cerc ca o număratoare
//liniară în mai multe siruri mici, fiecare cu n copii, obținând un sir mare cu
//p copii (p fiind multiplu de n). Număratoarea se termină atunci când al n-lea
//copil (dintr-un sir mic) primește ciocolata (astfel, următorul copil care ar
//trebui să primească ciocolata va fi un al k-lea copil din următorul sir mic),
//deci p trebuie să fie și multiplu de k. Asadar, p = cmmmc(n, k). Dintre cei
//p copii, au primit ciocolată exact p / k copii, deci copiii fără ciocolata sunt
//în număr de nr = n - p/k = n - cmmmc(n,k)/k = n - (n*k/cmmdc(n,k))/k = n - n/cmmdc(n,k)
int degustareCiocolata(int n, int k){
    return n - n / cmmdc(n, k);
}

```

REZOLVARE – Subiect B.2.: Numere magice.....15 puncte

```

#include <iostream>
using namespace std;

// se construiește vectorul de apariții a cifrelor în baza p pentru un număr x
// se determină pe rând cifrele în baza q ale numărului x
// dacă cifra curentă nu apare în reprezentarea în baza p atunci numărul x nu este magic
// altfel se incrementează valoarea corespunzătoare cifrei în vectorul de cifre
// dacă în vectorul de cifre a rămas valoarea 1 pentru anumite cifre, acele cifre apar în
// reprezentarea în baza p și nu apar în reprezentarea în baza q, deci numărul nu este magic

bool nrMagic(int x, int p, int q){
    //verifică dacă x este magic în raport cu bazele p și q
    int cifre[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
    int copie = x;
    while (copie != 0){ //stabilim de câte ori apare cifra în baza p
        int uc = copie % p; //uc - ultima cifră (în baza p)
        cifre[uc]++;
        copie = copie / p;
    }
    copie = x;
    while (copie != 0){ //determinăm cifrele lui x în baza q
        int uc = copie % q;
        if (cifre[uc] == 0) //dacă cifra curentă (în baza q) nu e folosită în baza p
            return false;
        cifre[uc]++;
        copie = copie / q;
    }
    for (int i = 0; i < 10; i++){
        if (cifre[i] == 1) //dacă cifra i e folosită în baza p, dar nu e folosită în baza q
            return false;
    }
    return true;
}

```

```

void sirNrMagice(int p, int q, int n, int &k, int sir[]){
    k = 0;
    for (int i = 1; i < n; i++){
        if (nrMagic(i, p, q))
            sir[k++] = i;
    }
}

```

REZOLVARE – Subiect B.3.: Căutare 10 puncte

Linia 5 trebuie modificată în: returnează ((x[n] = val) and cautare(x, n - 1, val))

- 1: Subalgoritm cautare(x, n, val):
- 2: Dacă n = 0 atunci
- 3: returnează x[0] = val
- 4: altfel
- 5: returnează ((x[n] = val) and cautare(x, n - 1, val))
- 6: SfDacă
- 7: SfSubalgoritm

REZOLVARE - Subiect B. 4. Cifra de control..... 10 puncte

- 1: Subalgoritm cifraDeControl(x):
- 2: Returnează x mod 9
- 3: SfSubalgoritm