

CLUJ 2018
Model 2

Subiectul A (30 puncte)

1. (5p) Fie următorul subalgoritm:

```
Subalgoritm f(a):
    Dacă a = 0 atunci
        returnează a + f(a - 1)
    altfel
        returnează 0
    SfDacă
SfSubalgoritm
```

Care din afirmațiile de mai jos este falsă?

- ☒ a. f este un subalgoritm definit recursiv
☐ b. dacă a este negativ, subalgoritmul întoarce 0
☐ c. valoarea calculată de f este $a * (a + 1) / 4$
☐ d. subalgoritmul calculează suma numerelor mai mici sau egale cu a
☐ e. apelul f(-5) intră în ciclul infinit

2. (5p) Se dă următorul subalgoritm

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

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

```
x ← 4;
y ← 3;
z ← f(x, y);
```

- ☒ a. de 4 ori
☐ b. de 3 ori
☐ c. de o infinitate de ori
☐ d. niciodată
☐ e. o dată

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

- ☒ a. 7
☐ b. 80
☐ c. 81
☐ d. 78
☐ e. 2

4. (5p) O matrice cu 8 linii, formată doar din elemente 0 și 1, are următoarele trei proprietăți:

- a. prima linie conține un singur element cu valoarea 1,
b. linia j conține de două ori mai multe elemente nenule decât linia j - 1, pentru orice $j \in \{2, 3, \dots, 8\}$,
c. ultima linie conține un singur element cu valoarea 0.

Care este numărul total de elemente cu valoarea 0 din matrice?

- ☒ a. 777
☐ b. 769
☐ c. 528

- d. nu există o astfel de matrice
e. 1

5. (5p) Se dau 3 șiruri a, b, c cu n, m, respectiv k elemente și următorii subalgoritmi:

```
Subalgoritm F1(x, 1):
    s ← 0
    Pentru i ← 1, 1 execută
        s ← s + x[i]
    SfPentru
    returnează s
SfSubalgoritm

Subalgoritm F2(n1, n2):
    returnează n1 + n2
SfSubalgoritm
```

Care dintre următoarele instrucțiuni sunt corecte în cazul existenței a 3 șiruri (a, b, c) cu câte n, m și, respectiv, k numere naturale:

- ☒ a. $F2(F1(a, n), F1(b, m)), k$
☐ b. $val = F1(c, k) + F2(F1(b, m), F1(a, n))$
☐ c. $val = F1(c, k) + F2(F1(a, m), b, n)$
☐ d. $F2(F2(F1(a, n), F1(b, m)), F1(c, k))$
☐ e. $val = F1(k, c) + F2(F1(m, b), F1(n, a))$

6. (5p) Se dă următorul subalgoritm:

```
Subalgoritm fc(a, s):
    k ← 0
    Pentru i ← 1, lungime(s) execută:
        k ← k + a
    SfPentru
    returnează k
SfSubalgoritm
```

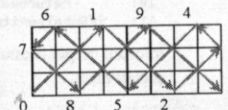
Precizați care dintre secvențele de instrucțiuni de mai jos vor produce afișarea numărului 75?
Observație: s-a presupus că indexarea șirurilor începe de la 1.

- a. $nr = fc("ana", 25)$
afișare(nr)
☒ b. $nr = fc(25, "ana")$
afișare(nr)
☐ c. $afișare(fc(25, "ana"))$
☐ d. nu există un astfel de apel
☐ e. $afișare(fc("ana", 25))$

Subiectul B (60 puncte)

1. Rază (25 puncte)

Avem la dispoziție un chenar dreptunghiular format din oglinzi. O rază de lumină pornește din colțul stânga jos al dreptunghiului sub un unghi de 45° față de latura de jos a dreptunghiului și lovește latura de sus sau latura din dreapta. Aici se reflectă (pornește spre o altă latură tot sub un unghi de 45° față de latura de care s-a lovit). Își continuă drumul până când ajunge într-un colț al dreptunghiului.



Scrieți un subalgoritm care calculează de câte ori (**nrSchimb**) raza își schimbă direcția de mers până când se oprește într-un colț. Punctul de pornire nu se numără. Parametri de intrare ai subalgoritmului sunt lungimea ($1 < a < 10\,000$) și lățimea ($1 < b < 10\,000$) dreptunghiului, iar **nrSchimb** va fi parametru de ieșire ($a, b, nrSchimb \in \mathbb{N}$).

Exemplu 1: dacă $a = 8$ și $b = 3$, atunci **nrSchimb** = 9.

Exemplu 2: dacă $a = 8$ și $b = 4$, atunci **nrSchimb** = 1.

2. Viruși (15 puncte)

În cadrul unui experiment, o populație de n ($3 \leq n \leq 1000$) viruși poate evolua astfel:

- dacă la începutul unei ore populația este formată dintr-un număr *par* de viruși, atunci la sfârșitul orei populația va fi mai mică cu 50%;
- dacă la începutul unei ore populația este formată dintr-un număr *impar* de viruși, atunci la sfârșitul orei populația de viruși va crește cu 1 virus;
- dacă la sfârșitul unei ore populația este formată dintr-un număr de viruși *strict mai mic decât un număr critic de supraviețuire*, atunci populația dispare.

Scrieți un subalgoritm care determină numărul de ore, notat *nrOre*, necesar distrugerii unei populații inițiale de n viruși, cunoscând numărul critic de supraviețuire k ($2 \leq k < n$). Parametrii de intrare sunt n și k , iar *nrOre* va fi parametru de ieșire.

Exemplu: dacă $n = 11$ și $k = 3$, populația se distruge în *nrOre* = 5.

3. Sortare (10 puncte)

Se dă următorul subalgoritm:

```

1: Subalgoritm sortare(a, n):
2:   Dacă  $n > 0$  atunci
3:     sortare(a, n - 1)
4:      $x \leftarrow a[n]$ 
5:      $j \leftarrow n - 1$ 
6:     CâtTimp ( $j \geq 0$  and  $a[j] > x$ ) execută:
7:        $j \leftarrow j - 1$ 
8:     SfCâtTimp
9:      $a[j + 1] \leftarrow x$ 
10:    SfDacă
11:  SfSubalgoritm

```

Ce instrucțiune/instrucțiuni trebuie adăugate, și unde, astfel încât în urma apelului subalgoritmului *sortare(a, n)* șirul a cu n elemente numere naturale să fie sortat?

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:      $s \leftarrow 0$ 
4:     CâtTimp  $x > 0$  execută:
5:        $s \leftarrow s + x \text{ MOD } 10$  {  $x \text{ mod } 10$  calculează restul împărțirii lui  $x$  la 10 }
6:        $x \leftarrow x \text{ DIV } 10$  {  $x \text{ div } 10$  calculează câtul împărțirii lui  $x$  la 10 }
7:     SfCâtTimp
8:      $x \leftarrow s$ 
9:   SfCâtTimp
10:  returnează  $x$ 
11: SfSubalgoritm

```

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

Notă:

- Toate subiectele sunt obligatorii.
- Rezolvările trebuie scrise detaliat pe foile de examen (ciomele nu se iau în considerare).
- Se acordă 10 puncte din oficiu.
- Timul efectiv de lucru este de 3 ore.

BAREM

OFICIU 10 puncte

SUBIECTUL A 30 puncte

- Răspunsurile b, c, d 5 puncte
- Răspunsul a 5 puncte
- Răspunsul a 5 puncte
- Răspunsul a 5 puncte
- Răspunsul b 5 puncte
- Răspunsurile b și c 5 puncte

SUBIECTUL B 60 puncte

B. 1. Rază 25 puncte

- determinarea corectă a valorii *nrSchimb* bazată pe utilizarea *cmmdc(a, b)* 25 puncte
- cmmdc(a, b)* (sau *mmmc(a, b)*) 10 puncte
- calculul valorii *nrSchimb* 15 puncte
- determinarea corectă a valorii *nrSchimb* cu alt algoritm corect (simulare) 15 puncte

B. 2. Viruși 15 puncte

- Rezolvare iterativă sau recursivă 10 puncte
- Calcul corect (populația moare la sfârșitul unei ore) 5 puncte

B. 3. Sortare 10 puncte

- identificare instrucțiune($a[j + 1] \leftarrow a[j]$) 5 puncte
- inserarea instrucțiunii între liniile 6 și 7 5 puncte

B. 4. Cifra de control 10 puncte

- cifra de control a unui număr poate fi calculată ca *nr mod 9* 10 puncte

REZOLVARE

REZOLVARE – Subiect B.1.: Rază..... 25 puncte

```
//determina cmmdc a 2 numere a si b
int cmmdc(int a, int b){
    if ((a == b) && (a != 0))
        return 1;
    if (a * b == 0)
        return a + b;
    while (a != b)
        if (a > b)
            a -= b;
        else
            b -= a;
    return a;
}

// calcularea numărului de schimbări de direcție a razei
int raza(int a, int b){
    int d = cmmdc(a, b);
    return b / d + a / d - 2;
}
```

REZOLVARE – Subiect B.2.: Virusi..... 15 puncte

```
//determina nr de ore necesar distrugerii unei populatii cu n virusi,
//pentru un nr critic de supravietuire k
int virusi(int n, int k){
    bool distrus = (n < k);
    int nrOre = 0;
    while (!distrus){
        if (n % 2 == 0) //daca avem nr par de virusi, inumatatim populatia
            n = n / 2;
        else //daca avem nr impar de virusi, marim populatia cu un virus
            n = n + 1;
        nrOre = nrOre + 1;
        distrus = (n < k); //verificam daca populatia dispare
    }
    return nrOre;
}
```

REZOLVARE – Subiect B.3.: Sortare..... 10 puncte

Între linia 6 și 7 trebuie inserată instrucțiunea $a[j + 1] \leftarrow a[j]$, subalgoritmul devenind

```
1: Subalgoritm sortare(a, n):
2:   Dacă  $n > 0$  atunci
3:      $f(a, n - 1)$ 
4:      $x \leftarrow a[n]$ 
5:      $j \leftarrow n - 1$ 
6:     Cât timp ( $j \geq 0$  and  $a[j] > x$ ) execută:
7:        $a[j + 1] \leftarrow a[j]$ 
8:        $j \leftarrow j - 1$ 
9:     SfCâtTimp
10:     $a[j + 1] \leftarrow x$ 
11:   SfDacă
12:   SfSubalgoritm
```

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

```
1: Subalgoritm cifraDeControl(x):
2:   Returnează  $x \bmod 9$ 
3:   SfSubalgoritm
```