CLUJ 2018

Model 2

UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Concurs Mate-Info - model

Proba scrisă la Informatică

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 afirmatiile de mai jos este falsă?

a. *f* este un subalgoritm definit recursiv
b. daca *a* este negativ, subalgoritmul întoarce 0
c. valoarea calculată de *f* este *a* * (*a* + 1) / 4
d. subalgoritmul calculează suma numerelor naturale mai mici sau egale cu *a*
e. apelul *f*(-5) intră în ciclu 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 apeleaza functia *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 şirurile de lungime $l \in \{1, 2\}$ formate din litere din mulțimea $\{a, b, c, d, e\}$. Câte dintre aceste şiruri au elementele ordonate strict crescător şi un număr par de vocale? (*a* şi *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 şi 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 $j \in \{2, 3, \ldots, 8\}$,
   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 şiruri *a*, *b*, *c* cu *n*, *m*, respectiv *k* elemente şi următorii subalgoritmi:

```
Subalgoritm F1(x, l):
    s ← 0
    Pentru i ← 1, l 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 şiruri (*a*, *b*, *c*) cu câte *n*, *m* şi, respectiv, *k* numere naturale:

a. F2(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:

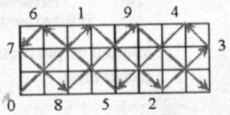
```
Subalgoritm fc(a, s):
    k = 0
    Pentru i ← 1, lungime(s) execută:
        k = k + a
    SfPentru
    returnează k
SfSubalgoritm
```

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

a. nr = fc("ana", 25)
   afişare(nr)
b. nr = fc(25, "ana")
   afişare(nr)
c. afişare(fc(25, "ana"))
d. nu există un astfel de apel
e. afişare(fc("ana", 25))

Subiectul B (60 puncte)

1. Rază (25 puncte)



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

Scrieți un subalgoritm care calculează de câte ori (***nrSchimb***) raza îşi schimbă direcția de mers până când se opreşte într-un colț. Punctul de pornire nu se numără. Parametri de intrare ai subalgoritmului sunt lungimea ($1 < a < 10\,000$) şi lățimea ($1 < b < 10\,000$) dreptunghiului, iar ***nrSchimb*** va fi parametru de ieşire (*a*, *b*, ***nrSchimb*** $\in \mathbb{N}$).
Exemplu 1: dacă $a = 8$ şi $b = 3$, atunci ***nrSchimb*** = 9.
Exemplu 2: dacă $a = 8$ şi $b = 4$, atunci ***nrSchimb*** = 1.

2. Viruşi (15 puncte)

În cadrul unui experiment, o populaţie de n ($3 \le n \le 1\,000$) viruşi poate evolua astfel:

a. dacă la începutul unei ore populaţia este formată dintr-un număr *par* de viruşi, atunci la sfârşitul orei populaţia va fi mai mică cu 50%;

b. dacă la începutul unei ore populaţia este formată dintr-un număr *impar* de viruşi, atunci la sfârşitul orei populaţia de viruşi va creşte cu 1 virus;

c. dacă la sfârşitul unei ore populaţia este formată dintr-un număr de viruşi *strict mai mic decât un număr critic de supravieţuire*, atunci populaţia dispare.

Scrieţi un subalgoritm care determină numărul de ore, notat ***nrOre***, necesar distrugerii unei populaţii iniţiale de n viruşi, cunoscând numărul critic de supravieţuire k ($2 \le k < n$). Parametrii de intrare sunt n şi k, iar ***nrOre*** va fi parametru de ieşire.

Exemplu: dacă $n = 11$ şi $k = 3$, populaţia 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 ← a[n]
5:        j ← n - 1
6:        CâtTimp (j >= 0 and a[j] > x) execută:
7:              j ← j - 1
8:        SfCâtTimp
9:        a[j + 1] ← x
10:    SfDacă
11:  SfSubalgoritm
```

Ce instrucţiune/instrucţiuni trebuie adăugate, şi unde, astfel încât în urma apelului subalgoritmului *sortare*(***a***, ***n***) şirul ***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 ← 0
4:          CâtTimp x > 0 execută:
5:              s ← s + x MOD 10 { x mod 10 calculează restul împărţirii lui x la 10}
6:              x ← x DIV 10     { x div 10 calculează câtul împărţirii lui x la 10}
7:          SfCâtTimp
8:          x ← s
9:       SfCâtTimp
10:      returnează x
11:  SfSubalgoritm
```

Înlocuiţi corpul acestui subalgoritm cu maxim 2 instrucţiuni astfel încât subalgoritmul rezultat să aibă acelaşi 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. Răspunsurile b, c, d ..... **5 puncte**

A. 2. Răspunsul a ..... **5 puncte**

A. 3. Răspunsul a ..... **5 puncte**

A. 4. Răspunsul a ..... **5 puncte**

A. 5. Răspunsul b ..... **5 puncte**

A. 6. Răspunsurile b şi c ..... **5 puncte**

SUBIECTUL B ..... **60 puncte**

B. 1. Rază ..... **25 puncte**

V1: determinarea corectă a valorii ***nrSchimb*** bazată pe utilizarea *cmmdc*(***a***, ***b***) ..... 25 puncte

– *cmmdc*(***a***, ***b***) (sau *cmmmc*(***a***,***b***)) ..... 10 puncte

– calculul valorii ***nrSchimb*** ..... 15 puncte

V2: determinarea corectă a valorii ***nrSchimb*** cu alt algoritm corect (simulare) ..... 15 puncte

B. 2. Viruşi ..... **15 puncte**

– Rezolvare iterativă sau recursivă ..... 10 puncte

– Calcul corect (populaţia moare la sfârşitul unei ore) ..... 5 puncte

B. 3. Sortare ..... **10 puncte**

– identificare instrucţiune(`a[j + 1] ← a[j]`) ..... 5 puncte

– inserarea instrucţiunii între liniile 6 şi 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 virsui, injumatatim populatia
			n = n / 2;
		else			//daca avem nr impar de virsui, 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] ← a[j]`, subalgoritmul devenind

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

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

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