BABEŞ-BOLYAI UNIVERSITY
FACULTY OF MATHEMATICS AND COMPUTER SCIENCE

Admission exam - model

Written Exam for Computer Science

Subject A (30 points)

1. **(5p)** An integer data type represented using ***x*** bits (***x*** is a strictly positive natural number) can hold integer values within the following interval:
   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)** Given the following subalgorithm

```
Subalgorithm f(a, b):
       If a > 1 then
              return b * f(a - 1, b)
       else
              return b * f(a + 1, b)
       EndIf
EndSubalgorithm
```

   State how many times function *f* will be called within the following code snippet:

```
x ← 4
y ← 3
z ← f(x, y)
```

   a. 4 times
   b. 3 times
   c. infinite number of times
   d. never
   e. once

3. **(5p)** Let ***x*** be an integer variable that holds the smallest non-zero natural number that is a multiple of 36 and that is divisible with all prime numbers smaller than 10. Which of the following expressions holds true:
   a. (x < 1000) and ((x*x*x) mod 1000 = 0)
   b. (x mod 100 = 0) or (x div 100 = 0)
   c. (x > 1000) && (x mod 7 = 0)
   d. ((x*x) div 16) mod 2 = 1
   e. ((x*x) div 16) mod 2 = 0

4. **(5p)** Consider all arrays having length $\boldsymbol{l} \in \{1, 2, 3\}$ built using letters from the set {*a*, *b*, *c*, *d*, *e*}. How many of them are sorted strictly decreasing and have an odd number of vowels? (*a* and *e* are vowels)
   a. 14
   b. 7
   c. 81
   d. 78
   e. 0

5. **(5p)** Given the following code snippet

```
void positiveNumbers(int m,int a[],
int &n, int b[]){
    n = 0;
    for(int i = 0; i < n; i++){
      if (a[i] > 0){
         n = n + 1;
         b[n] = a[i];
      }
    }
}
```

```
procedure positiveNumbers (m:integer;
a:array; var n:integer; var b:array)
  Begin
    n := 0;
    for i := 1 to n do
      if (a[i] > 0) then
        begin
          n := n + 1;
          b[n] := a[i];
        end;
  End;
```

What is the result of calling *positiveNumbers*(***k***,***x***,***l***,***y)*** for ***k***=4, array ***x***=(-1,2,-3,4), ***l*** = -1 and the empty array ***y*** = ().

a. l = 3 and y=[2, 4];
b. l = 0 and y=[2, 4];
c. l = 0 and y=[];
d. Depends on the value of *k*
e. Compile-time error

6. **(5p)** Given the following subalgorithm:

```
Subalgorithm SA6(a):
       If a < 50 then
              If a mod 3 = 0 then
                     return SA6(2 * a - 3)
              else
                     return SA6(2 * a - 1)
              EndIf
       else
              return a
       EndIf
EndSubalgorithm
```

For which value of parameter *a* will the subalgorithm return 61?

a. 16
b. 61
c. 4
d. 31
e. 51

<u>Subject B</u> (60 points)

1. Chocolate Tasting (25 points)

A marketing company is advertising a new brand of chocolate and intends to distribute chocolate samples to ***n*** ($10 \leq n \leq 10\,000\,000$) children who are standing in a circle. The employees know that distributing samples to each child would cost too much money. Therefore, they decide to distribute a sample to each ***k***-th ($0 < k < n$) child. They count the chidren standing in the circle, and hand out a sample to each ***k***-th child. When the count reaches the last child, it continues with the first one, and so on. All chidren are counted, even if they have already received chocolate. Counting stops when a chocolate sample should be given to a child who already received one.

Write a subalgorithm that determines the number of children (***nr***) who will not receive a chocolate sample. Input parameters are natural numbers ***n*** and ***k***, and the output parameter is the natural number ***nr***.

Example 1: if ***n*** = 12 and ***k*** = 9, then ***nr*** = 8 (the first, second, fourth, fifth, seventh, eighth, tenth, and eleventh child do not receive a sample).

Exemple 2: if ***n*** = 15 and ***k*** = 7, then ***nr*** = 0 (each child receives a chocolate sample).

2. Magic numbers (15 points)

Let us consider two natural numbers ***p*** and ***q*** ($2 \leq p \leq 10$, $2 \leq q \leq 10$). A natural number is *magic* when the set of digits used for writing its representation in base ***p*** is identical to the set of digits used for writing its representation in base ***q***. For example, for ***p*** = 9 and ***q*** = 7, $(31)_{10}$ is a m*agic* number because $(34)_9 = (43)_7$, and for ***p*** = 3 and ***q*** = 9, $(9)_{10}$ is a *magic* number because $(100)_3 = (10)_9$.

Write a subalgorithm that for given bases ***p*** and ***q*** determines the array ***x*** containing all *magic* numbers that are stricly positive and strictly smaller than a given natural number ***n*** ($1 < n \leq 10000$). The input parameters for the subalgorithm are ***p*** and ***q*** (the two bases) and the value ***n***. The output parameters are the array ***x*** and its length ***k***.
Example: if ***p*** = 9, ***q*** = 7 and ***n*** = 500, array ***x*** will contain ***k*** = 11 elements: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

3. Search (10 points)

Given the following subalgorithm:

```
1:    Subalgoritm search(x, n, val):
2:          If n = 0 then
3:                return (x[0] = val)
4:          else
5:                return search(x, n - 1, val)
6:          EndIf
7:    EndSubalgorithm
```

What code must be added and where so that the function above determines whether element *val* is contained within the ***n*** element array ***x*** (with ***n*** a strictly positive natural number)?

4. Control digit (10 points)

Consider the following subalgorithm that determines the control digit of a natural number that has at least 2 digits.

```
 1:   Subalgorithm controlDigit(x):
 2:       While x > 9 execute:
 3:         s ← 0
 4:         While x > 0 execute:
 5:             s ← s + x MOD 10 { x mod 10 is the remainder of dividing x by 10}
 6:             x ← x DIV 10    { x div 10 is the quotient of dividing x by 10}
 7:         EndWhile
 8:         x ← s
 9:       EndWhile
10:       return x
11:   EndSubalgorithm
```

Replace the body of this subalgorithm with maximum 2 instructions so that the subalgorithm has the same effect.

Note:

1. All subjects are mandatory.
2. Solutions must be written on the exam sheets in detailed form (drafts are not graded).
3. Default 10 points.
4. Working time is 3 hours.

Evaluation

Granted points ..... **10 points**

SUBJECT A ..... **30 points**

A. 1. Answers b,c ..... **5 points**

A. 2. Answer c ..... **5 points**

A. 3. Answers c,d ..... **5 points**

A. 4. Answer a ..... **5 points**

A. 5. Answer c ..... **5 points**

A. 6. Answers a, b, d ..... **5 points**

SUBJECT B ..... **60 points**

B. 1. Chocolate Tasting ..... **25 points**

- V1: correct determination of ***nr*** (by using ***nr*** = ***n*** – ***n***/*gcd*(***n***, ***k***) ..... 25 points
- V2: correct determination of ***nr*** (simulation, circular list)..... 15 points

B. 2. Magic numbers..... **15 points**

– property of *magic number*

- V1: by using characteristic vectors associated to the sets of number digits corresponding to two representations (base ***p*** and base ***q***, respectively) ..... 10 points
- V2: other versions..... maxim 5 points

– constraction of vector ***x***..... 5 points

B. 3. Search ..... **10 points**

– identification of condition`(x[n] = val)` ..... 5 points

– return the truth value of composed condition from row 5..... 5 points

B. 4. Control digit ..... **10 points**

– control digit can be computed as ***nr*** `mod 9`..... 10 points

Solving – Subject B.1.: Chocolate tasting.....25 points

```
#include <iostream>
using namespace std;
/*******************************************************************************
Subiectul I.1. Degustare de ciocolata
*******************************************************************************/
//greatest common divisor of 2 numbers a and 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;
}

int degustareCiocolata(int n, int k){
	return n - n / cmmdc(n, k);
}
```

Solving – Subject B.2.: Magic numbers.....15 points

```
#include <iostream>
using namespace std;

bool nrMagic(int x, int p, int q){
   int cifre[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
   int copie = x;
   while (copie != 0){     //stabilim multimea cifrelor lui x in baza p
      int uc = copie % p;  //uc - ultima cifra (in baza p)
      cifre[uc] = 1;
      copie = copie / p;
   }
   copie = x;
   while (copie != 0){     //determinam cifrele lui x in baza q
      int uc = copie % q;
      if (cifre[uc] == 0)  //daca cifra curenta (in baza q) nu e folosita in baza p
         return false;
      cifre[uc]++;
      copie = copie / q;
   }
   for (int i = 0; i < 10; i++){
      if (cifre[i] == 1)   //daca cifra i e folosita in baza p, dar nu e folosita in 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;
   }
}
```

Solving – Subject B.3.: Search ..... **10 points**

Row 5 must be changed: `return ((x[n] = val) and search(x, n - 1, val))`

```
1:    Subalgorithm search(x, n, val):
2:          If n = 0 then
3:                return x[0] = val
4:          else
5:                return ((x[n] = val) and search(x, n - 1, val))
6:          EndIf
7:    EndSubalgorithm
```

Solving - Subject B. 4. Control digit ..... **10 points**

```
1:   Subalgorithm controlDigit(x):
2:       Return x mod 9
3:   ENdSubalgorithm
```