UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Aufnahmeprüfung Mate-Info - Modell

Schriftliche Prüfung in Informatik

Aufgabe A (30 Punkte)

1. **(5p)** Ein Ganzzahl-Datentyp (Integer), der auf x Bits dargestellt wird (x ist eine natürliche streng positive Zahl) kann als Werte ganze Zahlen aus dem folgenden Intervall speichern:
   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)** Gegeben sei folgender Subalgorithmus:

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

   Gebe an, wie viele Male die Funktion *f* in folgenden Anweisungsblock aufgerufen wurde:

```
x ← 4
y ← 3
z ← f(x, y)
```

   a. 4-mal
   b. 3-mal
   c. unendlich Male
   d. keinmal
   e. einmal

3. **(5p)** Sei x eine ganzzahlige Variable, welche die kleinste natürliche Zahl ungleich von 0 enthält, die zusätzlich Vielfaches von 36 und teilbar durch alle primen Zahlen kleiner als 10 ist. Gebe an, welche der folgenden Ausdrücke wahr sind.
   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)** Seien alle Zeichenfolgen mit Länge $\boldsymbol{l} \in \{1, 2, 3\}$ bestehend aus den Buchstaben in der Menge {*a*, *b*, *c*, *d*, *e*}. Wie viele dieser Zeichenfolgen haben die Elemente in streng absteigender Reihenfolge geordnet und haben zusätzlich eine ungerade Anzahl von Vokalen. (*a* und *e* sind Vokale)
   a. 14
   b. 7
   c. 81
   d. 78
   e. 0

5. **(5p)** Gegeben seien folgende Anweisungsblöcke:

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

```
procedure positiveZahlen(m:integer; a:sir;
var n:integer; var b:sir)
  Begin
    n := 0;
    for i := 1 to n do
      if (a[i] > 0) then
```

```
            b[n] = a[i];
        }
    }
}
```

```
        begin
         n := n + 1;
         b[n] := a[i];
        end;
End;
```

Welches ist das Ergebnis der Ausführung für den Aufruf *positiveZahlen*(***k***,***x***,***l***,***y)*** mit k=4, ***x***=(-1,2,-3,4), ***l*** = -1 und dem leeren Array ***y*** = ().

a. l = 3 und y=[2, 4];
b. l = 0 und y=[2, 4];
c. l = 0 und y=[];
d. Hängt vom Wert von k ab
e. Kompillierungsfehler

6. **(5p)** Sei der folgende Subalgorithmus:

```
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
```

Für welche der folgende Werte des Parameters *a* wird das Subalgorithmus 61 zurückgeben?

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

<u>Aufgabe B</u> (60 Punkte)

1. Schokoladenverkostung (25 Punkte)

Eine Werbefirma wirbt eine neue Schokoladensorte und plant Schokoladenproben zu ***n*** ($10 \leq n \leq 10000000$) Kindern, die in einem Kreis sitzen, zu verteilen. Die Angestellten der Firma merken, dass die Verteilung der Proben zu allen Kindern sehr teuer ist. Folglich, entscheiden sie Proben zu jedem ***k***-te ($0 < k < n$) Kind aus den ***n*** Kindern zu verteilen. Man zählt also die Kinder in Schritte mit dem Schritt ***k*** (wenn man zu dem letzten Kind ankommt, dann geht das Zählen weiter mit dem ersten Kind und so weiter). Beim Zählen werden alle Kinder berücksichtigt, egal ob das Kind Schokolade bekommen hat oder nicht. Das Zählen *hört genau dann auf wenn eine Schokoladenprobe zu einem Kind verteilt werden muss, das schon Schokolade bekommen hat*.

Schreibe einen Subalgorithmus, der die Anzahl der Kinder (***nr***) bestimmt, welche keine Schokoladenprobe erhalten. Die Eingabeparameter des Subalgorithmus sind die natürliche Zahlen ***n*** und ***k***, und der Rückgabewert (Ausgabeparameter) wird die natürliche Zahl ***nr*** sein.

Beispiel 1: wenn ***n*** = 12 und ***k*** = 9, dann ***nr*** = 8 (das zweite, vierte, fünfte, siebente, achte, zehnte und elfte Kind erhalten keine Schokolade).

Beispiel 2: wenn ***n*** = 15 und ***k*** = 7, dann ***nr*** = 0 (alle Kinder erhalten Schokolade).

2. Magische Zahlen (15 Punkte)

Seien ***p*** und ***q*** zwei natürliche Zahlen ($2 \leq p \leq 10$, $2 \leq q \leq 10$). Eine natürliche Zahl heißt *magisch*, wenn die Menge der Ziffern, die benutzt wird um die Zahl in dem Zahlensystem mit Basis ***p*** darzustellen, identisch ist mit der Menge der Ziffern, die benutzt wird um die Zahl in dem Zahlensystem mit Basis ***q*** darzustellen. Zum Beispiel, für ***p*** = 9 und ***q*** = 7, ist $(31)_{10}$ eine *magische* Zahl, da $(34)_9 = (43)_7$, und für ***p*** = 3 und ***q*** = 9, ist $(9)_{10}$ eine magische Zahl, da $(100)_3 = (10)_9$.

Schreibe ein Subalgorithmus, der für zwei gegebene Basis ***p*** und ***q*** eine Sequenz ***x*** erstellt mit allen *magischen* Zahlen streng größer als 0 und streng kleiner als eine gegebene natürliche Zahl ***n*** ($1 < n \leq 10\,000$). Die Eingabeparameter des Subalgorithmus sind ***p*** und ***q*** (die zwei Basis) und der Wert ***n***. Die Rückwerte sind die Sequenz ***x*** und die Länge ***k*** der Sequenz ***x.***

Beispiel: wenn ***p*** = 9, ***q*** = 7 și ***n*** = 500, dann wird die Sequenz ***x*** ***k*** = 11 Elemente haben: (1, 2, 3, 4, 5, 6, 31, 99, 198, 248, 297).

3. Suche (10 Punkte)

Gegebei sei folgender Subalgorithmus:

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

Welche Anweisung oder Anweisungen müssen hinzugefügt werden und wo müssen diese hinzugefügt werden, sodass, nach dem Aufruf, die Funktion *suche* feststellt, ob der Element ***val*** in dem Array ***x*** mit ***n*** Elemente enthalten ist oder nicht (***n*** ist eine natürliche Zahl streng positiv)?

4. Kontrollziffer (10 Punkte)

Gegeben sei folgender Subalgorithmus für das Bestimmen der Kontrollziffer für eine Zahl mit wenigstens zwei Ziffern.

```
 1:  Subalgorithm kontrollziffer(x):
 2:      While x > 9 execute:
 3:        s ← 0
 4:        While x > 0 execute:
 5:            s ← s + x MOD 10 { x mod 10 berechnet der Rest der Division von x
                                   durch 10}
 6:            x ← x DIV 10    { x div 10 berechnet den Quotient der Division von x
                                  durch 10}
 7:        EndWhile
 8:        x ← s
 9:      EndWhile
10:      return x
11:  EndSubalgorithm
```

Ersetze den Anweisungsblock dieses Subalgorithmus mit höchstens 2 Anweisungen sodass der neue Subalgorithmus dieselbe Wirkung hat.

Bemerkung:

1. Alle Aufgaben sind verpflichtend.
2. Die Lösungen müssen detailiert auf die Prüfungsblätter geschrieben werden (Schmierblätter werden nicht berücksichtigt)
3. Die Anfangspunkteanzahl ist 10 (din oficiu).
4. Die Bearbeitungszeit ist 3 Stunden.

BEWERTUNG

ANFANGSPUNKTEANZAHL (OFICIU) ..... **10 Punkte**

AUFGABE A ..... **30 Punkte**

A. 1. Antwort b,c ..... **5 Punkte**

A. 2. Antwort c ..... **5 Punkte**

A. 3. Antwort c,d ..... **5 Punkte**

A. 4. Antwort a ..... **5 Punkte**

A. 5. Antwort c ..... **5 Punkte**

A. 6. Antwort a, b, d ..... **5 Punkte**

AUFGABE B ..... **60 Punkte**

B. 1. Schokoladenverkostung ..... **25 Punkte**

- V1: die korrekte Bestimmung des Wertes ***nr*** (mit der Formel ***nr*** = ***n*** – ***n***/*cmmdc*(***n***, ***k***)) 25 Punkte
- V2: die korrekte Bestimmung des Wertes ***nr*** (durch Simulation, zirkuläre Liste) .....15 Punkte

B. 2. Magische Zahlen ..... **15 Punkte**

– Die Überprüfung der Eigenschaft einer *magischen Zahl*

- V1: mithilfe der Identität der zwei characteristischen Vektoren für die Mengen der Ziffern der Zahl in den zwei Zahlensystemen (mit Basis ***p***, beziehungsweise Basis ***q)***.....10 Punkte
- V2: andere Algorithmus-Varianten, die korrekt sind, aber nicht so effizient....maxim 5 Punkte

– Die Aufbau der Sequenz ***x*** .....5 Punkte

B. 3. Suche ..... **10 Punkte**

– Die Bestimmung der Bedingung `(x[n] = val)` .....5 Punkte

– Die Rückgabe des Wahrheitswertes für die zusammengesetzte Bedingung aus Linie 5 .....5 Punkte

B. 4. Kontrollziffer ..... **10 Punkte**

– Die Kontrollziffer einer Zahl kann als ***nr*** `mod 9` berechnet werden .....10 Punkte

LÖSUNG – Aufgabe B.1.: Schokoladenverkostung.....25 Punkte

```
#include <iostream>
using namespace std;
/*******************************************************************************
Aufgabe I.1. Schokoladenverkostung
*******************************************************************************/
//berechnet und gibt zurück das cmmdc (kleinste gemeinsame Vielfache) zweier natürlichen
//Zahlen a und 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;
}

// bestimmt und gibt zurück die Anzahl der Kinder (aus den n Kindern), die keine Schokolade
// erhalten, wenn man jedes k-te Kind zählt. Man kann das Zählen im Kreis als lineares Zählen
// in mehreren kleinen Sequenzen betrachten, wobei jede Sequenz n Kinder enthält. Dann kriegt
// man am Ende eine lange Sequenz mit p Kinder, wobei p Vielfache von n ist. Das Zählen wird
// beendet, wenn das n-te Kind (aus einer kleinen Sequenz) Schokolade erhaltet (so ist das
// nächste Kind, das Schokolade erhalten muss, das k-te Kind aus der nächsten Sequenz), also p
// muss auch Vielfache von k sein. Deshalb, p = cmmmc(n, k). Aus den p Kinder, haben genau
// p / k Kinder Schokolade erhalten, also die Anzahl der Kinder, die keine Schokolade erhalten
// haben, ist nr = n - p/k = n - cmmmc(n,k)/k = n - (n*k/cmmdc(n,k))/k = n - n/cmmdc(n,k)

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

LÖSUNG – Aufgabe B.2.: Magische Zahlen.....15 Punkte

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

// man erstellt den charakteristichen Vektor der Ziffern der Zahl x zur Basis p
// man bestimmt der Reihe nach die Ziffern der Zahl x zur Basis q
//	wenn die aktuelle Ziffer nicht in der Zahlendarstellung zur Basis p vorkommt, dann ist
//		x nicht magisch
//	ansonsten wird der entsprechende Wert in dem Ziffernvektor inkrementiert
// wenn in dem Ziffernvektor ein Wert 1 vorhanden ist, dann heißt das, dass die entsprechenden
// Ziffern in der Zahlendarstellung zur Basis p, aber nicht in der Zahlendarstellung zur Basis
// q vorkommen. Folglich ist die Zahl nicht magisch.


bool magischeZahl(int x, int p, int q){
   //überprüft ob x magisch ist in Bezug auf Basis p und q
   int ziffern[10] = { 0, 0, 0, 0, 0, 0, 0, 0, 0, 0 };
   int kopie = x;
   while (kopie != 0){     //man bestimmt die Menge der Ziffern der Zahl x zur Basis p
      int lz = kopie % p;  //lz – letzte Ziffer (zur Basis p)
      ziffern[lz] = 1;
      kopie = kopie / p;
   }
   kopie = x;
   while (kopie != 0){       //man bestimmt die Ziffern der Zahl x zur Basis q
      int lz = kopie % q;
      if (ziffern[lz] == 0)   //wenn die aktuelle Ziffer (zur Basis q) nicht in der
         return false;        //Zahlendarstellung zur Basis p vorkommt
      ziffern[lz]++;
      kopie = kopie / q;
```

```
   }
   for (int i = 0; i < 10; i++){
      if (ziffern[i] == 1)    //wenn die Ziffer i in der Zahlendarstellung zur Bais p, aber
         return false;        //nicht zur Basis q verwendet wird
   }
   return true;
}

void sequenzMagischeZahlen(int p, int q, int n, int &k, int seq[]){
   k = 0;
   for (int i = 1; i < n; i++){
      if (magischeZahl(i, p, q))
         seq[k++] = i;
   }
}
```

LÖSUNG – Aufgabe B.3.: Suche ..... 10 Punkte

Linie 5 muss folgendermaße geändert werden: `return ((x[n] = val) and suche(x, n - 1, val))`

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

LÖSUNG – Aufgabe B. 4. Kontrollziffer ..... 10 Punkte

```
1:   Subalgorihtm kontrollziffer(x):
2:       Return x mod 9
3:   SfSubalgorithm
```