UNIVERSITATEA BABEŞ-BOLYAI
FACULTATEA DE MATEMATICĂ ŞI INFORMATICĂ

Wettbewerb Mate-Info - Modell

Schriftliche Prüfung in Informatik

Aufgabe A (30 Punkte)

1. **(5p)** Sei der folgende Subalgorithmus:

```
Subalgorithm f(a):
       If a != 0 then
              return a + f(a - 1)
       else
              return 0
       EndIf
EndSubalgorithm
```

Welche der folgenden Aussagen ist falsch?

a. *f* ist ein rekursiver Subalgorithmus
b. falls a negativ ist, dann gibt der Subalgorithmus den Wert 0 zurück
c. der Wert, der von f zurückgegeben wird ist $\boldsymbol{a} * (\boldsymbol{a} + 1) / 4$
d. der Subalgorithmus berechnet die Summe der natürlichen Zahlen kleiner oder gleich ***a***
e. der Aufruf ***f***(-5) läuft unendlich (ciclu infinit).

2. **(5p)** Gegeben sei der folgende Subalgorithmus:

```
Subalgorithm f(a, b):
       If (a > 1) then
              return b * f(a - 1, b)
       else
              return 1
       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)** Seien alle Zeichenfolgen mit Länge $\boldsymbol{l} \in \{1, 2\}$ bestehend aus den Buchstaben in der Menge {*a, b, c, d, e*}. Wie viele dieser Zeichenfolgen haben die Elemente in streng steigender Reihenfolge geordnet und haben zusätzlich eine gerade Anzahl von Vokalen. (*a* und *e* sind Vokale)

a. 7
b. 80
c. 81
d. 78
e. 2

4. **(5p)** Eine Matrix mit 8 Linien, die nur die Werte 0 und 1 enthält, hat folgende drei Eigenschaften:
   a. die erste Linie enthält ein einziges Element mit Wert 1,
   b. Linie ***j*** enthält zweimal so viele Elemente verschieden von 0 wie Linie ***j***-1, für jede $\boldsymbol{j} \in \{2, 3, \ldots, 8\}$,
   c. die letzte Linie enthält ein einziges Element mit Wert 0

   Welche ist die Anzahl der Elemente mit Wert 0 aus der Matrix?

a. 777
b. 769
c. 528
d. Es gibt keine Matrix mit diesen Eigenschaften
e. 1

5. **(5p)** Gegeben seien 3 Arrays ***a***, ***b***, ***c*** mit ***n***, ***m***, beziehungsweise ***k*** Elemente und folgende Subalgorithmen:

```
Subalgorithm F1(x, l):
        s ← 0
        For i ← 1, l execute
                s ← s + x[i]
        EndFor
        return s
EndSubalgorithm
```

```
Subalgorithm F2(n1, n2):
        return n1 + n2
EndSubalgorithm
```

Welche der folgenden Anweisungen sind korrekt im Falle der Existenz der 3 Arrays (***a***, ***b***, ***c***) mit ***n***, ***m*** und, beziehungsweise, ***k*** natürlichen Zahlen:

```
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)** Gegeben sei der folgende Subalgorithmus:

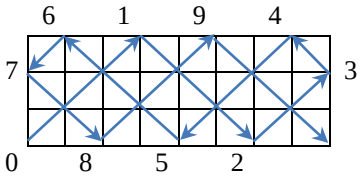
```
Subalgorithm fc(a, s):
        k = 0
        For i ← 1, länge(s) execute:
                k = k + a
        EndFor
        return k
EndSubalgorithm
```

Welche der folgenden Anweisungsblöcke wird die Zahl 75 ausdrucken?
Bemerkung: man nimmt an, dass die Indexierung des Arrays mit 1 beginnt.

a. nr = fc("ana", 25)
   print(nr)
b. nr = fc(25, "ana")
   print(nr)
c. print(fc(25, "ana"))
d. es gibt kein solches Aufruf
e. print(fc("ana", 25))

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

1. Lichtstrahl (25 Punkte)



Man hat Spiegeln in einer rechteckigen Form und ein Lichtstrahl in der unteren linken Ecke unter einem Winkel von 45^o bezüglich der unteren Kante des Rechtecks. Der Lichtstrahl trifft die obige oder die rechte Kante und reflektiert (der Lichtstrahl geht also weiter, wieder unter einem Winkel von 45^o bezüglich der Kante, die sie getroffen hat). Der Lichtstrahl verbreitet

sich so weiter, bis er zu einer Ecke des Rechtecks gelangt.

Schreibe einen Subalgorithmus, der die Anzahl (***nrÄnderungen***) berechnet, wie viele Male der Lichtstrahl seine Richtung ändert bis er zu dem Endpunkt gelangt (in einer Ecke des Rechtecks). Der Startpunkt wird nicht dazu gezählt. Die Eingabeparameter des Subalgorithmus sind die Länge ($1 < \boldsymbol{a} < 10\ 000$) und die Breite ($1 < \boldsymbol{b} < 10\ 000$) des Rechtecks, und der Rückgabewert ist ***nrÄnderungen*** (***a***, ***b***, ***nrSchimb*** $\in \mathbf{N}$).

Beispiel 1: falls $\boldsymbol{a} = 8$ und $\boldsymbol{b} = 3$, dann ***nrÄnderungen*** = 9.
Beispiel 2: falls $\boldsymbol{a} = 8$ und $\boldsymbol{b} = 4$, dann ***nrÄnderungen*** = 1.

2. Viren (15 Punkte)

In einem Experiment, kann sich eine Population von $\boldsymbol{n}$ ($3 \leq \boldsymbol{n} \leq 1\,000$) Viren folgendermaße entwickeln:

- **a.** falls am Anfang einer Stunde die Population aus einer *geraden* Anzahl von Viren besteht, dann wird die Population am Ende der Stunde um 50% kleiner sein;
- **b.** falls am Ende einer Stunde die Population aus einer un*geraden* Anzahl von Viren besteht, dann wird die Population am Ende der Stunde mit 1 Virus vergrößert;
- **c.** falls die Population am Ende einer Stunde aus einer Anzahl von Viren *streng kleiner als eine kritische Überlebensrate* besteht, dann stirbt die Population.

Schreibe einen Subalgorithmus, der die Anzahl von Stunden (***nrStunden)*** berechnet, in welchen eine Population von ***n*** Viren stirbt (die kritische Überlebensrate $\boldsymbol{k}$, $2 \leq \boldsymbol{k} < \boldsymbol{n}$, ist gegeben). Die Eingabeparameter sind ***n*** und ***k***, und der Rückgabewert ist ***nrStunden***.

Beispiel: wenn $\boldsymbol{n} = 11$ und $\boldsymbol{k} = 3$, dann stirbt die Population in ***nrStunden*** = 5 Stunden.

3. Sortieren (10 Punkte)

Gegeben sei der folgende Subalgorithmus:

```
1:   Subalgorithm sortieren(a, n):
2:     If n > 0 then
3:        sortieren (a, n - 1)
4:        x ← a[n]
5:        j ← n - 1
6:        While (j >= 0 and a[j] > x) execute:
7:              j ← j - 1
8:        EndWhile
9:        a[j + 1] ← x
10:    EndIf
11:  EndSubalgorithm
```

Welche Anweisung oder Anweisungen müssen hinzugefügt werden und wo müssen diese hinzugefügt werden, sodass, der Aufruf des Subalgorithmus, *sortieren*(***a***, ***n***), den Array ***a*** mit ***n*** Elementen (natürliche Zahlen) sortiert?

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 den 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, d ..... **5 Punkte**

A. 2. Antwort a ..... **5 Punkte**

A. 3. Antwort a ..... **5 Punkte**

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

A. 5. Antwort b ..... **5 Punkte**

A. 6. Antwort b und c ..... **5 Punkte**

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

B. 1. Lichstrahl ..... **25 Punkte**

V1: die korrekte Bestimmung des Wertes ***nrÄnderungen*** mithilfe von *cmmdc*(***a***, ***b***) .....25 Punkte

- *cmmdc*(***a***, ***b***) (oder *cmmmc*(***a***,***b***)) .....10 Punkte
- die Berechnung des Wertes ***nrÄnderungen*** .....15 Punkte

V2: die korrekte Bestimmung des Wertes ***nrÄnderungen*** mit einem anderen korrekten Algorithmus (Simulation) .....15 Punkte

B. 2. Viren ..... **15 Punkte**

- iterative oder rekursive Lösung .....10 Punkte
- korrekte Berechnung (die Population stirbt am Ende einer Stunde) .....5 Punkte

B. 3. Sortieren ..... **10 Punkte**

- Indetifizierung der Anweisung `(a[j + 1] ← a[j])` .....5 Punkte
- Einfügen der Anweisung zwischen Linie 6 und 7 .....5 Punkte

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

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

LÖSUNGEN

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

```
//man bestimmt cmmdc (größte gemeinsame Teiler) zweier 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 (a != b)
		if (a > b)
			a -= b;
		else
			b -= a;
	return a;
}

// man berechnet die Anzahl der Richtungsänderungen des Lichtstrahls
int lichtstrahl(int a, int b){
	int d = cmmdc(a, b);
	return b / d + a / d - 2;
}
```

LÖSUNG – Aufgabe B.2.: Viren ..... 15 Punkte

```
//bestimmt die Anzahl von Stunden, die nötig sind, damit eine Population von n Viren mit einer
//kritischen Überlebensrate k stirbt

int viren(int n, int k){
	bool stirbt = (n < k);
	int nrStunden = 0;
	while (!stirbt){
		if (n % 2 == 0)		//falls die Anzahl der Viren eine gerade Zahl ist, dann wird
			n = n / 2;	//die Population halbiert

		else			//falls die Anzahl der Viren eine ungerade Zahl ist, dann
			n = n + 1;	//wird die Population mit einem Virus vergrößert
		nrStunden = nrStunden + 1;
		stirbt = (n < k);	//man überprüft ob die Population stirbt
	}
	return nrStunden;
}
```

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

Zwischen der Linie 6 und 7 muss die Anweisung `a[j + 1] ← a[j]` eingefügt werden:

```
 1:   Subalgorithm sortieren(a, n):
 2:     If n > 0 then
 3:        f(a, n - 1)
 4:        x ← a[n]
 5:        j ← n - 1
 6:        While (j >= 0 and a[j] > x) execute:
 7:               a[j + 1] ← a[j]
 8:               j ← j - 1
 9:        EndWhile
10:        a[j + 1] ← x
11:     EndIf
12:   EndSubalgorithm
```

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

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