



**Lösungshinweise: Nur für die Hand der Lehrperson**

**Schriftliche Abiturprüfung 2015**

Fach: **Informatik**

Kurstyp: G-Kurs

Bearbeitungszeit: 3 Zeitstunden

Hilfsmittel: nicht-programmierbarer Taschenrechner

Seitenzahl: Die Lösungshinweise umfassen mit Deckblatt, Punkteverteilungstabelle und Bewertungstabelle 11 Seiten.



**1. Aufgabe (20 Punkte)**

**1.1 (9,5 Punkte) (Lehrplan: "Objektorientiertes Modellieren und Programmieren", "Strukturiertes Programmieren")**

**1.1.1 (3 Punkte)**

<b>TDreieck</b>
<code>x1, y1, x2, y2, x3, y3: real</code>
<code>gibSeitenlaenge(z: char): real;</code> <code>gibFlaecheninhalt(): real;</code> <code>gibGroessteSeitenlaenge(): real;</code> <code>gibMittlereSeitenlaenge(): real;</code> <code>gibKleinsteSeitenlaenge(): real;</code> <code>istKongruent(d:TDreieck): boolean;</code>

**1.1.2 (3 Punkte)**

Java-Code:

```
double gibSeitenlaenge(char z) {  
    if (z == 'c')  
        return Math.sqrt(((x1-x2)*(x1-x2)+(y1-y2)*(y1-y2)));  
    else  
        if (z == 'b')  
            return Math.sqrt((x1-x3)*(x1-x3)+(y1-y3)*(y1-y3));  
        else  
            if (z == 'a')  
                return Math.sqrt((x2-x3)*(x2-x3)+(y2-y3)*(y2-y3));  
            else return -1;  
}
```

PASCAL/Delphi-Code:

```
function TDreieck.gibSeitenlaenge (z: char):real;
begin
    if z='c'
    then result:=sqrt(((x1-x2)*(x1-x2))+((y1-y2)*(y1-y2)))
    else
    if z='b'
    then result:=sqrt(((x1-x3)*(x1-x3))+((y1-y3)*(y1-y3)))
    else
    if z='a'
    then result:=sqrt(((x2-x3)*(x2-x3))+((y2-y3)*(y2-y3)))
    else result:=-1;
end;
```

### 1.1.3 (3,5 Punkte)

Java-Code:

```
boolean istKongruent(TDreieck d) {
    double max1= gibGroessteSeitenlaenge();
    double mittel= gibMittlereSeitenlaenge();
    double min1= gibKleinsteSeitenlaenge();

    double max2= d.gibGroessteSeitenlaenge();
    double mitte2= d.gibMittlereSeitenlaenge();
    double min2= d.gibKleinsteSeitenlaenge();

    return (max1==max2)&&(mittel==mitte2)&&(min1==min2);
}
```

PASCAL/Delphi-Code:

```
function TDreieck.istKongruent(d: TDreieck):boolean;
var max1,max2,mittel,mitte2,min1,min2:real;
begin
    max1:= gibGroessteSeitenlaenge();
    mittel:= gibMittlereSeitenlaenge();
    min1:= gibKleinsteSeitenlaenge();

    max2:= d.gibGroessteSeitenlaenge();
    mitte2:= d.gibMittlereSeitenlaenge();
    min2:= d.gibKleinsteSeitenlaenge();

    result:=(max1=max2)and(mittel=mitte2)and(min1=min2);
end;
```

**1.2 (10,5 Punkte) (Lehrplan: "Suchen und Sortieren")****1.2.1 (3 Punkte)**

18	7	5	41	32	0
7	18	5	41	32	0
5	7	18	41	32	0
5	7	18	41	32	0
5	7	18	32	41	0
0	5	7	18	32	41
-	-	-	-	-	-

**1.2.2 (1,5 Punkte)**

Die Laufzeitklasse von Insertionsort ist  $O(n^2)$ . Das Mergesortverfahren hat ein besseres Laufzeitverhalten:  $O(n \cdot \log n)$ .

**1.2.3 (3 Punkte)**

18	7	5	41	32	0
0	18	7	41	32	5
0	5	18	41	32	7
0	5	7	41	32	18
0	5	7	18	41	32
0	5	7	18	32	41
-	-	-	-	-	-

**1.2.4 (3 Punkte)**

1. Durchgang: n-1 Vergleiche
2. Durchgang: n-2 Vergleiche
3. Durchgang: n-3 Vergleiche
4. Durchgang: .....

Insgesamt:  $(n-1) + (n-2) + (n-3) + \dots + 1 = \sum_{i=1}^{n-1} i = \frac{(n-1) \cdot n}{2} = \frac{n^2}{2} - \frac{n}{2} \rightarrow O(n^2)$

**2. Aufgabe (20 Punkte)**

**2.1 (6,5 Punkte) (Lehrplan: „Strukturiertes Programmieren“)**

**2.1.1 (1 Punkt)**

Die nächsten beiden Folgenglieder lauten:

$t_5 = 0110\ 1001\ 1001\ 0110$

$t_6 = 0110\ 1001\ 1001\ 0110\ 1001\ 0110\ 0110\ 1001$

**2.1.2 (3 Punkte)**

Java-Code:

```
public String thuemorseIt(int n){
    String t="0";
    for(int i=2; i<=n; i++){
        t=t + negiere(t);
    }
    return t;
}
```

PASCAL/Delphi-Code:

```
function thuemorseIt(n:integer):String;
var t:String;
    i:integer;
begin
    t:= '0';
    for i:=2 to n do
        t:=t + negiere(t);
    result:=t;
end;
```

**2.1.3 (2,5 Punkte)**

Java-Code:

```
public String thuemorseRek(int n){
    if (n == 1) return "0";
    else return thuemorseRek(n-1) + negiere(thuemorseRek(n-1));
}
```

PASCAL/Delphi-Code:

```
function thuemorseRek(n:integer):String;
begin
    if n = 1 then result:= '0'
    else result:=thuemorseRek(n-1) + negiere(thuemorseRek(n-1));
end;
```

## 2.2 (13,5 Punkte) (Lehrplan: „Automaten und formale Sprachen“)

### 2.2.1 (1,5 Punkte)

$t_1$ :  $S \rightarrow 0B \rightarrow 0$

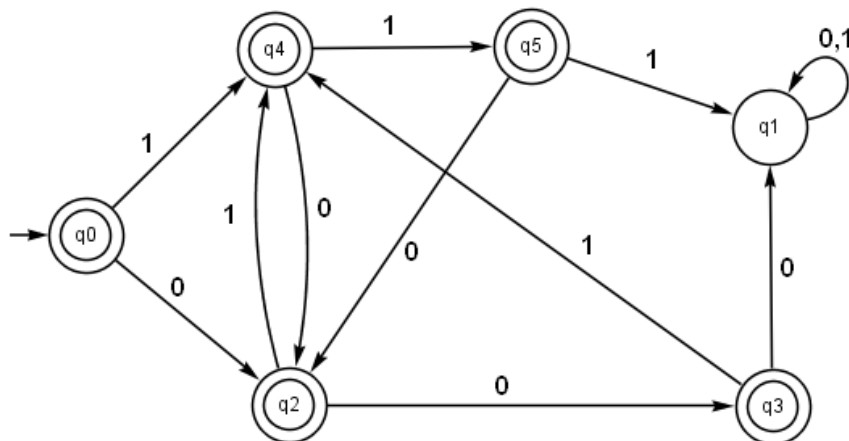
$t_2$ :  $S \rightarrow 0B \rightarrow 01A \rightarrow 01$

$t_3$ :  $S \rightarrow 0B \rightarrow 01A \rightarrow 011C \rightarrow 0110B \rightarrow 0110$

### 2.2.2 (4 Punkte)

DEA =  $(Q, \Sigma, \delta, q_0, F)$  mit

- $Q = \{q_0, q_1, q_2, q_3, q_4, q_5\}$
- $\Sigma = \{0, 1\}$
- $F = \{q_0, q_2, q_3, q_4, q_5\}$
- $\delta$ : siehe Zustandsübergangsgraph



Anmerkung: Der Zustand  $q_1$  ist nicht unbedingt erforderlich und kann weggelassen werden.

### 2.2.3 (2 Punkte)

Die regulären Ausdrücke beschreiben unterschiedliche Sprachen. Z.B. gilt  $0110 \notin L(\alpha)$ , aber  $0110 \in L(\beta)$

### 2.2.4 (2 Punkte)

Die von endlichen Automaten akzeptierten Sprachen sind äquivalent zu den regulären Sprachen. Da in Aufgabe 2.2.2 ein Automat angegeben wurde, welcher die gegebene Sprache akzeptiert, muss diese Sprache regulär sein und somit existiert ein regulärer Ausdruck, welcher diese Sprache beschreibt.

**2.2.5 (4 Punkte)**

Grammatik  $G = (N, T, P, S)$  mit

$N = \{S, A, B, C, D\}$

$T = \{0, 1\}$

$P = \{$

(1)	$S \rightarrow 1A1 \mid 0B0 \mid 1 \mid 0 \mid 11 \mid 00 \mid \varepsilon,$
(2)	$A \rightarrow 1C1 \mid 0B0 \mid 0 \mid 00,$
(3)	$B \rightarrow 0D0 \mid 1A1 \mid 1 \mid 11,$
(4)	$C \rightarrow 0B0 \mid 0 \mid 00,$
(5)	$D \rightarrow 1A1 \mid 1 \mid 11$

$\}$

Es handelt sich um keine reguläre Sprache, da beispielsweise die Produktionen  $S \rightarrow 1A1 \mid 0B0$  nicht durch reguläre Produktionen ersetzt werden können.

**3. Aufgabe (20 Punkte)**

**3.1 (10,5 Punkte) (Lehrplan: "Kryptographie")**

**3.1.1 (1 Punkte)**

Willi hat die Zahl  $e$  falsch gewählt. Diese muss teilerfremd zu  $\varphi(77) = 60$  sein.

**3.1.2 (4 Punkte)**

Bestimmung des privaten Schlüssels  $(d, 55)$  zum öffentlichen Schlüssel  $(7, 55)$ :

$$\text{Es gilt: } \varphi(55) = (p - 1) \cdot (q - 1) = 4 \cdot 10 = 40$$

Gesucht ist also ein  $d$  mit  $7 \cdot d - s \cdot 40 = 1$  für irgend ein  $s$ .

Anwendung des erweiterten Euklidischen Algorithmus:

$$40 = 5 \cdot 7 + 5$$

$$7 = 1 \cdot 5 + 2$$

$$5 = 2 \cdot 2 + 1$$

Damit ergibt sich:

$$1 = 5 - 2 \cdot 2$$

$$= 5 - 2 \cdot (7 - 1 \cdot 5)$$

$$= 3 \cdot 5 - 2 \cdot 7$$

$$= 3 \cdot (40 - 5 \cdot 7) - 2 \cdot 7$$

$$= 3 \cdot 40 - 15 \cdot 7 - 2 \cdot 7$$

$$= 3 \cdot 40 - 17 \cdot 7$$

Also ist  $d = -17 \equiv 23 \pmod{40}$ ; es gilt:  $1 \equiv 23 \cdot 7 \pmod{40}$

Der private Schlüssel lautet:  $(23, 55)$

**3.1.3 (2,5 Punkte)**

$$B \rightarrow 2 \quad 2^7 \pmod{55} = 18$$

$$E \rightarrow 5 \quad 5^7 \pmod{55} = 25$$

Eddie erhält die Zahlenfolge 18 25 25.

**3.1.4 (3 Punkte)**

Der Chef des Geheimdienstes hat Recht. Werden beim RSA-Verfahren die Zeichen des Klartextes einzeln verschlüsselt, so handelt es sich um ein monoalphabetisches Substitutionsverfahren. Ein solches kann leicht mittels einer Häufigkeitsanalyse gebrochen werden. Daher muss beim RSA-Verfahren blockweise chiffriert werden.



**3.2 (9,5 Punkte) (Lehrplan: "Funktionsweise von Computersystemen")****3.2.1 (3,5 Punkte)**

Speicherstelle 1	0	7	13	21	5	0
Speicherstelle 2	0	7	13	21	21	

Ausgabe: 21

Das Programm gibt die größte Zahl der Zahlenfolge aus.

**3.2.2 (6 Punkte)**

0	JMP	4	
1	DEF	0	gerade Zahl, die in jedem Schritt hinzuaddiert werden soll
2	DEF	0	Summe
3	DEF	0	n
4	INM	3	n wird eingelesen
5	LDA	1	
6	INC		
7	INC		Die Zahl 2 (4, ...) wird zur bisherigen Summe addiert
8	STA	1	
9	ADD	2	
10	STA	2	
11	LDA	3	Eingegebene Zahl wird mit zu addierender Zahl verglichen
12	SUB	1	
13	JZE	18	Akkueinhalt =0 -> Addition ist beendet
14	JNM	5	Akkueinhalt >0 -> es muss weiter addiert werden
15	LDA	2	Akkueinhalt <0 -> eingegebene Zahl war ungerade
16	SUB	1	es wurde einmal zu viel addiert; Ergebnis muss korrigiert werden
17	STA	2	
18	OUT	2	Ausgabe der Summe
19	END		

**Punkteverteilungstabellen:****Aufgabe 1:**

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
1.1	9,5	1,5	8	
1.2	10,5	1,5	6	3
	<b>20</b>	<b>3</b>	<b>14</b>	<b>3</b>
		<b>15%</b>	<b>70%</b>	<b>15%</b>

**Aufgabe 2:**

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
2.1	6,5	1	5,5	
2.2	13,5	5	6,5	2
	<b>20</b>	<b>6</b>	<b>12</b>	<b>2</b>
		<b>30%</b>	<b>60%</b>	<b>10%</b>

**Aufgabe 3:**

Teilaufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
3.1	10,5	1,5	6	3
3.2	9,5	1,5	7	1
	<b>20</b>	<b>3</b>	<b>13</b>	<b>4</b>
		<b>15%</b>	<b>65%</b>	<b>20%</b>

**Punkteverteilungstabelle (insgesamt):**

Aufgaben	Summe	Punkte in den Anforderungsbereichen		
		I	II	III
1	20	3	14	3
2	20	6	12	2
3	20	3	13	4
	<b>60</b>	<b>12</b>	<b>39</b>	<b>9</b>
		<b>20%</b>	<b>65%</b>	<b>15%</b>

**Bewertungstabelle**

<b>Prozent der maximal erreichbaren Rohpunktzahl</b>	<b>Note</b>	<b>Punktzahl</b>
ab 97% bis 100% der max. Punktzahl	sehr gut	15 P
ab 93% bis weniger als 97%		14 P
ab 90% bis weniger als 93%		13 P
ab 85% bis weniger als 90%	gut	12 P
ab 80% bis weniger als 85%		11 P
ab 75% bis weniger als 80%		10 P
ab 70% bis weniger als 75%	befriedigend	09 P
ab 65% bis weniger als 70%		08 P
ab 60% bis weniger als 65%		07 P
ab 55% bis weniger als 60%	ausreichend	06 P
ab 50% bis weniger als 55%		05 P
ab 45% bis weniger als 50%		04 P
ab 38% bis weniger als 45%	mangelhaft	03 P
ab 32% bis weniger als 38%		02 P
ab 25% bis weniger als 32%		01 P
weniger als 25% der max. Punktzahl	ungenügend	00 P

**- Ende der Lösungshinweise -**