Aufgabe 1.1 (27 Punkte)

Kreuzen Sie bitte an, ob folgende Behauptungen richtig oder falsch sind.

Bem.: Für jede richtige Antwort gibt es einen Punkt, für falsche wird ein Punkt abgezogen, keine Anwort ist punktneutral. Insgesamt wird diese Aufgabe mit mindestens 0 und höchstens 27 Punkten bewertet.

	richtig	falsch
Es gibt logische Aussagen, die weder falsch noch wahr sind.		
Das Multiplikandenregister des Klassischen Universalrechners ist direkt mit dem Speicher verbunden.		
Das Ergebnis einer Multiplikation im Klassischen Universalrechner wird im Multiplikandenregister abgelegt.		
In einem PC (Personal Computer) sind wie beim Klassischen Universalrechner Speicherwerk und Rechenwerk strikt getrennt.		
Logische Aussagen sind immer falsch oder wahr.		
Die Wortbreite eines Rechners wird im wesentlichen von den Eigenschaften der CPU bestimmt.		
Negation ist die Implikation der Tautologie.		
Die Anzahl der Bits in einem Byte hängt von der Wortbreite des Rechners ab.		
Zu jeder aussagenlogischen Formel existiert eine äquivalente Formel, die als Junktoren nur die Konjunktion und Äquivalenz enthält.		
Wenn man statt der üblichen Darstellung von negativen und positiven Zahlen nur noch positive Zahlen darstellt, kann man den Wertebereich der darstellbaren Zahlen verdoppeln.		
Es gibt logische Aussagen, die entweder falsch oder wahr sind.		
Das Leitwerk des Klassischen Universalrechners steuert bei einem unbedingten Sprung über das Befehlsregister die Auswahl des nächsten Befehls.		
Es gibt keine logische Aussage, die immer falsch oder immer wahr ist.		
Die boolesche Algebra gestattet es, logische Probleme durch Rechnen zu lösen.		
Der klassische Universalrechner arbeitet analog.		
Der klassische Universalrechner hat getrennte Einheiten für die Speicherung von Daten und Programmen.		
Das Leitwerk des klassischen Universalrechners ist programmierbar.		
Die Darstellung der Werte "0" und "1" im klassischen Universalrechner erfolgt durch Belegen der Steuerleitungen mit "0V" bzw. "5V".		

	richtig	falsch
Die Speicherkapazität einer 9cm (3,5´´) High-Density Diskette reicht nicht aus, um eine schwarz-weiße Zeichnung mit 300dpi Auslösung zu speichern.		
Zwei Zahlen kann man durch geeignete Zusammenschaltung ausreichend vieler Halbaddierer multiplizieren.		
Rechner stellen Zeichenketten in Form von Zahlenfolgen in ihrem Arbeitsspeicher dar.		
Wenn man statt der üblichen Darstellung von negativen und positiven Zahlen nur noch positive Zahlen darstellt, kann man die Anzahl der darstellbaren Zahlen verdoppeln.		
Im Gegensatz zum PC (Personal Computer) hat der Klassische Universalrechner keine Erweiterungssteckplätze.		
Das Leitwerk eines Klassischen Universalrechners enthält neben dem Befehlsregister nur noch den Befehlszähler und Funktionsentschlüsslung.		
Die Erhöhung des Befehlszählers des Klassischen Universalrechners um Eins wird durch die ALU (Arithmetisch-Logische-Einheit) durchgeführt.		
Die Befehlsholphase beim Klassischen Universalrechner wird ausgelassen, wenn der nächste Befehl der "STOP"-Befehl ist.		
Die Befehlsausführungsphase beim Klassischen Universalrechner wird ausgelassen, wenn der auszuführende Befehl der "STOP"-Befehl ist.		

Aufgabe 1. 2 (9 Punkte)

Geben Sie für folgende Sachverhalte eine oder mehrere boolesche Funktionen an:

1. Ein Eierkocher produziert hartgekochte Eier, wenn Eier und genügend Wasser eingefüllt und das Gerät eingeschaltet wurde.
2. Es gibt Hitzefrei, wenn es drei Tage hintereinander um 10 Uhr mindestens 25° warm war.
3. Boomerang: Wenn man es wegwirft, und es kommt nicht zurück, dann war es keiner.

Aufgabe 1. 3 (16 Punkte)

Geben Sie die Symboldarstellung für folgende boolesche Funktionen an:

1. $y = (\neg x_1 \vee x_2 \wedge \neg(x_1 \vee x_4 \wedge \neg(x_3 \vee x_2)))$
2. $y = (\neg x_2 \vee \neg(x_3 \wedge x_1) \vee x_1) \wedge x_2$
3. $y = \neg x_3 \wedge (((x_1 \vee x_1) \wedge (x_2 \vee x_1) \vee x_1) \wedge \neg(x_2 \wedge x_4))$
4. $y = (x_2 \vee \neg x_1) \wedge \neg(x_1 \vee x_3)$

Aufgabe 2. 1 (10 Punkte)

Kreuzen Sie bitte an, ob folgende Behauptungen richtig oder falsch sind.

Bem.: Für jede richtige Antwort gibt es einen Punkt, für falsche wird ein Punkt abgezogen, keine Anwort ist punktneutral. Insgesamt wird diese Aufgabe mit mindestens 0 und höchstens 10 Punkten bewertet.

	richtig	falsch
Effektivität eines Algorithmus bedeutet, daß er möglichst wenig Speicherplatz und Rechenzeit benötigt.		
Manche Algorithmen liefern ein Ergebnis.		
Ein Programm, das eine Endlosschleife enthält, kann ein Ergebnis liefern .		
Jedes Struktogramm beschreibt einen Algorithmus.		
Jeder Algorithmus kann durch ein Struktogramm beschrieben werden.		
Algorithmen, die durch ein Struktogramm beschrieben sind, enthalten keine Endlosschleifen.		
Für Struktogramme bestehen besonders strenge Vorschriften, was den Inhalt der Struktogramme betrifft.		
Struktogramme gestatten mindestens eine grobe Beschreibung der Datenstrukturen, indem der Programmierer entsprechende Kommentaren ins Struktogramm einfügt.		
Es gibt Programme, die nie ein richtiges Ergebnis liefern.		
Es gibt Programme, die nie ein falsches Ergebnis liefern.		

Aufgabe 2. 2 (15 Punkte)

Formulieren Sie einen Algorithmus, der folgende Aufgabe löst:

Der alphabetisch kleinste und größte Buchstaben einer Folge von Eingabezeichen soll ermittelt werden. Das Programm soll über eine Prozedur verfügen, die den nächsten Buchstaben einliest. Dabei sollen alle anderen Zeichen (d.h. Ziffern, Sonderzeichen, Leerzeichen etc.) überlesen werden. Das Zeichen "*" soll das Ende der Eingabe markieren und ausgegeben werden, wenn keines der Zeichen ein Buchstabe war.

Geben Sie den Algorithmus als Struktogramm oder in einer weit verbreiteten Programmiersprache an.

Kommentieren Sie den Algorithmus und geben Sie an, welche Funktion die verwendeten Variablen und Konstanten haben.

Aufgabe 3. 1 (16 Punkte)

Geben Sie präzise an, was das folgende Programm ausgibt. Dokumentieren Sie die Berechnung mit einem Trace (Werteverlaufsprotokoll).


```
program beispiel(output);
var i:integer;
procedure p(var j:integer);
begin
i:=i+1;
j:=i+5;
end;

procedure q;
var i,k:integer;
begin
i:=5;
p(k); write(i:5,k:5);
p(i);write(i:5);
end;

begin
i:=1;
q; write(i:5);
end.
```

Aufgabe 3. 2 (15 Punkte)

Schreiben Sie ein Assemblerprogramm, das das Maximum beliebig vieler Zahlen ermittelt. Das Endergebnis soll im Speicher abgelegt werden und am Ende ausgegeben werden. Der Benutzer soll die Möglichkeit haben, eine beliebig lange Zahlenfolge einzugeben, das Ende der Eingabe soll mit der Eingabe der Zahl Null signalisiert werden.

Gehen Sie davon aus, daß der Benutzer nur positive Zahlen einschl. der Null eingibt.

Es stehen folgende Assemblerbefehle zur Verfügung:

Befehl	Bedeutung
HALT	Hält das Programm an
IN	Liest einen Wert in den Akkumulator ein
OUT	Gibt den Wert des Akkumalators aus
STORE	n Speichert den Wert des Akkumulators in Speicherzelle n
LOAD	n Liest den Wert der Speicherzelle n in das Multiplikandenregister
JMP	n Der nächste Befehl steht in Speicherzelle n
JEZ	n Springe , falls der Akkumulator gleich Null ist
JNZ	n Springe , falls der Akkumulator ungleich Null ist
JGZ	n Springe , falls der Akkumulator grössergleich Null ist
JLZ	n Springe , falls der Akkumulator kleinergleich Null ist
CLEAR	Setzt den Inhalt des Akkumulators auf Null.
ADD SUB MUL DIV	Arithmetische Operatoren, der erste Operator ist der Akkumulator, der zweite das Multiplikandenregister, das Ergebnis steht im Akkumulator. Beispiel: SUB entspricht Akkumulator:= Akkumulator - Multiplikandenregister.

Aufgabe 3. 3 (17 Punkte)

Erstellen Sie eine Entscheidungstabelle, die folgenden Sachverhalt widerspiegelt:

- Am Beginn des Spiels sind n Hölzer vorhanden, n beliebig.
- Die Spieler A und B ziehen abwechselnd, Spieler A beginnt.
- Bei jedem Zug müssen mindestens 1 und maximal 3 Hölzer entfernt werden.
- Es können nicht mehr Hölzer entfernt werden, als vorhanden sind.
- Kann ein Spieler kein Holz mehr nehmen, so hat er verloren.

Füllen Sie nur die Spalten aus, die unbedingt notwendig sind.

Verwenden Sie nicht mehr als 6 Bedingungen und 10 Aktionen.

Bem.: Es kommt nicht darauf an, eine Gewinnstrategie zu entwickeln.

Aufgabe 3. 4 (15 Punkte)

Die angegebenen Schleifen sind daraufhin zu überprüfen, ob und unter welchen Umständen die Schleife endlos durchlaufen wird. Geben Sie präzise an, welche Bedingungen dies sind!

1. ...
```
While n <>0 do
Begin
...
n:=n+2;
...
End;
...
```

2. ...
```
While i >0 Do
Begin
...
i:=i div 2;
...
End;
...
```

3. ...
```
While i < n Do
Begin
...
i:=i+1;
...
End;
...
```


4. ...
```
r:=0;
Repeat r:=r+0.1;
...
Until r = 10;
...
```

5. ...
```
While i>0 Do n:=2*n;
...
```

Aufgabe 3. 5 (10 Punkte)

Erläutern sind folgende Begriffe im Zusammenhang mit OLE (Object Linking and Embedding):

- OLE
- OLE Container
- Embedding
- Linking
- Visual Editing