

Vorname:	
Nachname:	
Matrikelnummer:	

## Prüfung – Informationstechnik

Wintersemester 2011/2012

9. März 2012

Bitte legen Sie Ihren Lichtbildausweis bereit.

Sie haben für die Bearbeitung der Klausur 120 Minuten Zeit.

Diese Prüfung enthält **22** nummerierte Seiten inkl. Deckblatt.

**Bitte prüfen Sie die Vollständigkeit Ihres Exemplars!**

Bitte nicht mit rot oder grün schreibenden Stiften oder Bleistift ausfüllen!

Diesen Teil nicht ausfüllen.

Aufgabe	ZS	MO	BS	RK	DB	$\Sigma$	Note:
erreichte Punkte							
erzielbare Punkte	<b>40</b>	<b>50</b>	<b>60</b>	<b>55</b>	<b>35</b>	<b>240</b>	



Vorname, Name

Matrikelnummer

**Aufgabe ZS: Zahlensysteme und logische Schaltungen**

*Aufgabe ZS:  
40 Punkte*

Punkte

- a) Überführen Sie die unten angegebenen Zahlen in die jeweils anderen Zahlensysteme. *Hinweis: Achten Sie genau auf die jeweils angegebene Basis!*

1 ( 1111100 )<sub>2</sub> = ( ..... )<sub>10</sub> = ( ..... )<sub>7</sub>

2 ( 45,625 )<sub>10</sub> = ( ..... )<sub>2</sub>

- b) Unten sehen Sie eine an die IEEE 754 angelehnte Gleitkommazahl. Stellen Sie die kodierte Gleitkommazahl im Dezimalsystem dar.

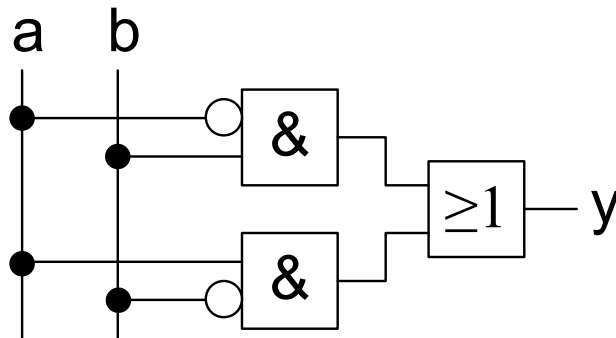
<b>1</b>	<b>10100</b>	<b>1100 0110 0000</b>
<b>V</b>	<b>biased Exponent e (5 Bits)</b>	<b>Mantisse (12 Bits)</b>



Vorname, Name

Matrikelnummer

Gegeben ist die folgende Schaltung:



Punkte

- c) Stellen Sie für die oben angegebene Schaltung die Wahrheitstabelle des Ausgangs  $y$  in Abhängigkeit der Eingänge  $a$  und  $b$  auf.

*Hinweis: Die freigelassenen Spaltennamen (erste Zeile) der Wahrheitstabelle können als Hilfsmittel für Zwischenergebnisse genutzt werden, werden jedoch nicht bewertet.*

a	b			y
0	0			
0	1			
1	0			
1	1			

- d) Erstellen Sie die *KNF (Konjunktive Normalform)* aus der Wahrheitstabelle, für die Ausgangsfunktion  $y$ .



Vorname, Name

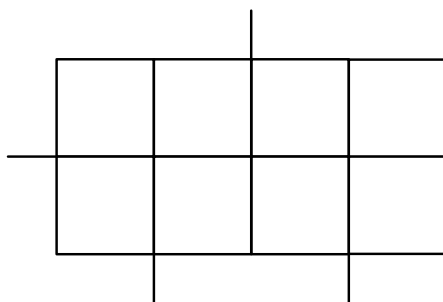
Matrikelnummer

Punkte

e) Gegeben ist folgende Wahrheitstabelle:

<b>a</b>	<b>b</b>	<b>c</b>	<b>y</b>
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

Erstellen Sie mit Hilfe des *KV-Diagramms* die minimierte Ausgangsfunktion für  $y_{\min}$  in *DNF (Diskunktive Normalform)* und schreiben Sie die minimierte Form in boolescher Algebra auf. Beschriften Sie hierfür das KV-Diagramm entsprechend und zeichnen Sie die Schleifen, die für die Minimierung der Funktion benötigt werden ein in das KV-Diagramm ein.

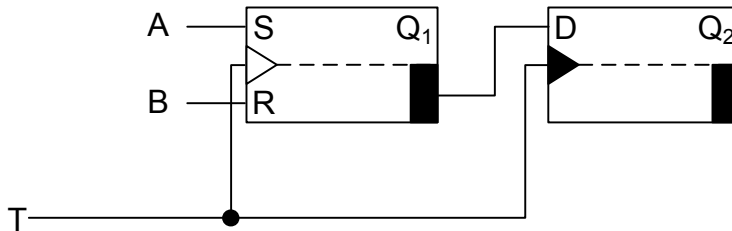




Vorname, Name

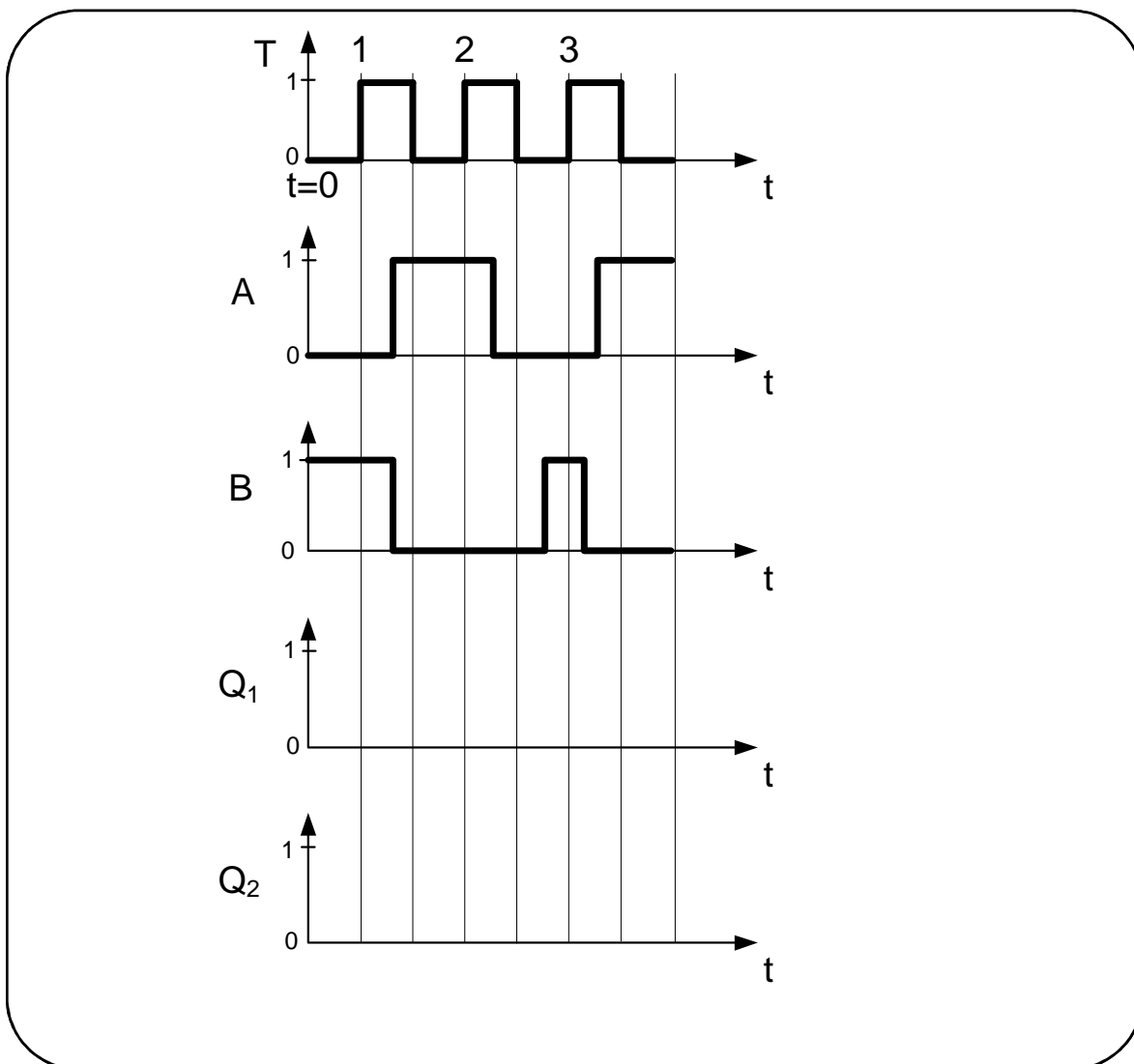
Matrikelnummer

f) Gegeben sei die folgende Master-Slave FlipFlop-Schaltung:



Bei  $t = 0$  seien Master- und Slave FlipFlop im Zustand  $Q_1 = Q_2 = 0$ .

Analysieren Sie die Schaltung, indem Sie die zeitlichen Verläufe für  $Q_1$  und  $Q_2$  in die vorgegebenen Koordinatensysteme eintragen. Die Signallaufzeiten können dabei vernachlässigt werden.



Punkte



Vorname, Name

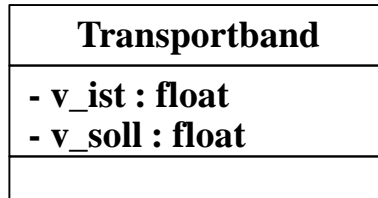
Matrikelnummer

**Aufgabe MO: Modellierung**

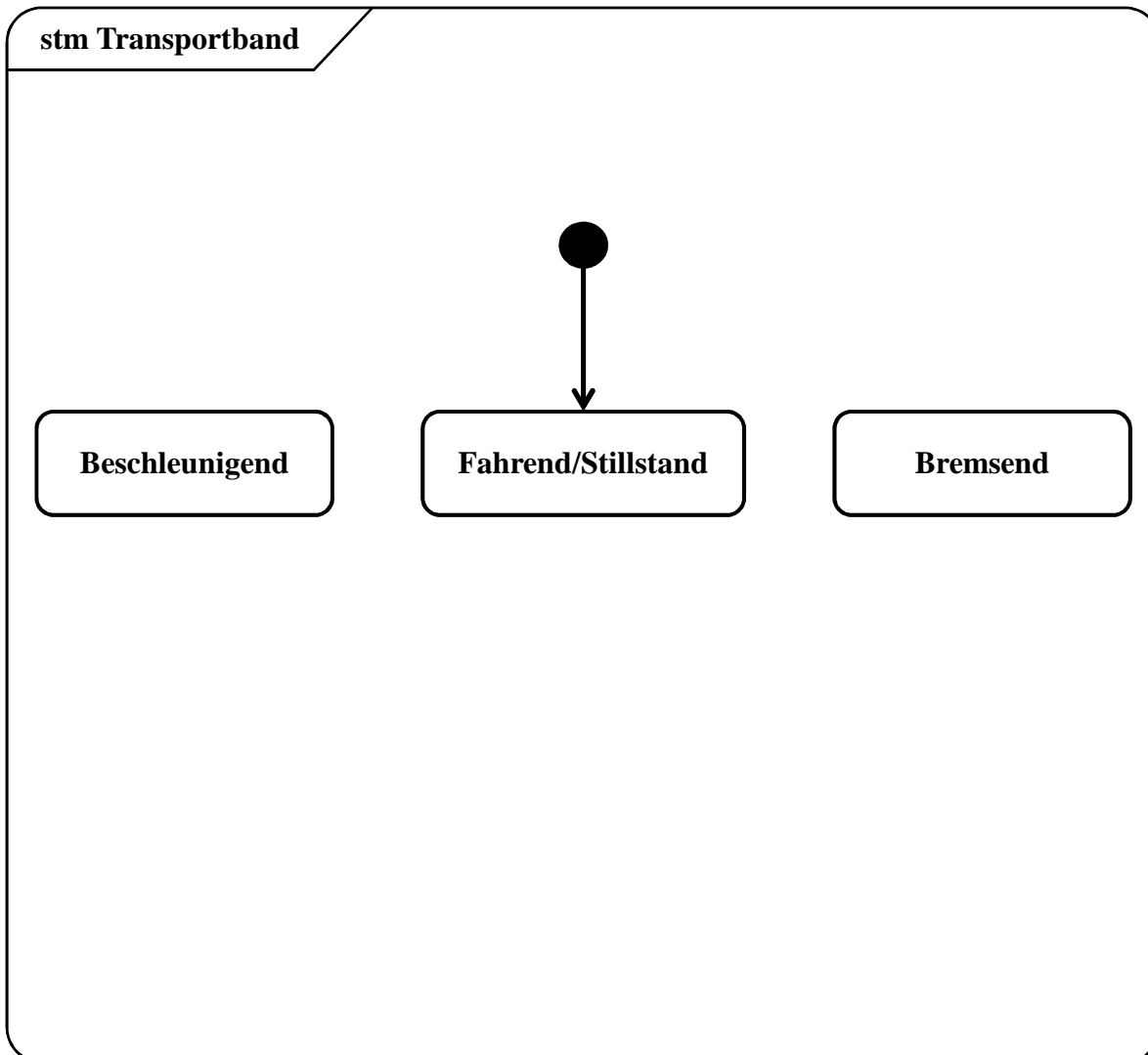
*Aufgabe MO:  
50Punkte*

Punkte

- a) Gegeben sei eine Klasse zur Beschreibung des Transportbands eines Transportsystems:



Die aktuelle Istgeschwindigkeit des Transportbands sei über das Attribut „v\_ist : float“ modelliert und die aktuelle Sollgeschwindigkeit über das Attribut „v\_soll : float“. *Das Transportband beschleunigt bzw. bremst in Abhängigkeit dieser Attribute.* Die Implementierung dieses Verhaltens soll durch das UML Zustandsdiagramm im Lösungsfeld beschrieben werden. Zeichnen Sie dazu in dem unten im Lösungsfeld alle nötigen Transitionen zwischen den angegebenen Zuständen zur Darstellung der beschriebenen Funktionalität ein. Zeichnen Sie dabei ebenfalls die jeweiligen Transitionsbedingungen in Abhängigkeit der Attribute der Klasse des Transportbands ein.



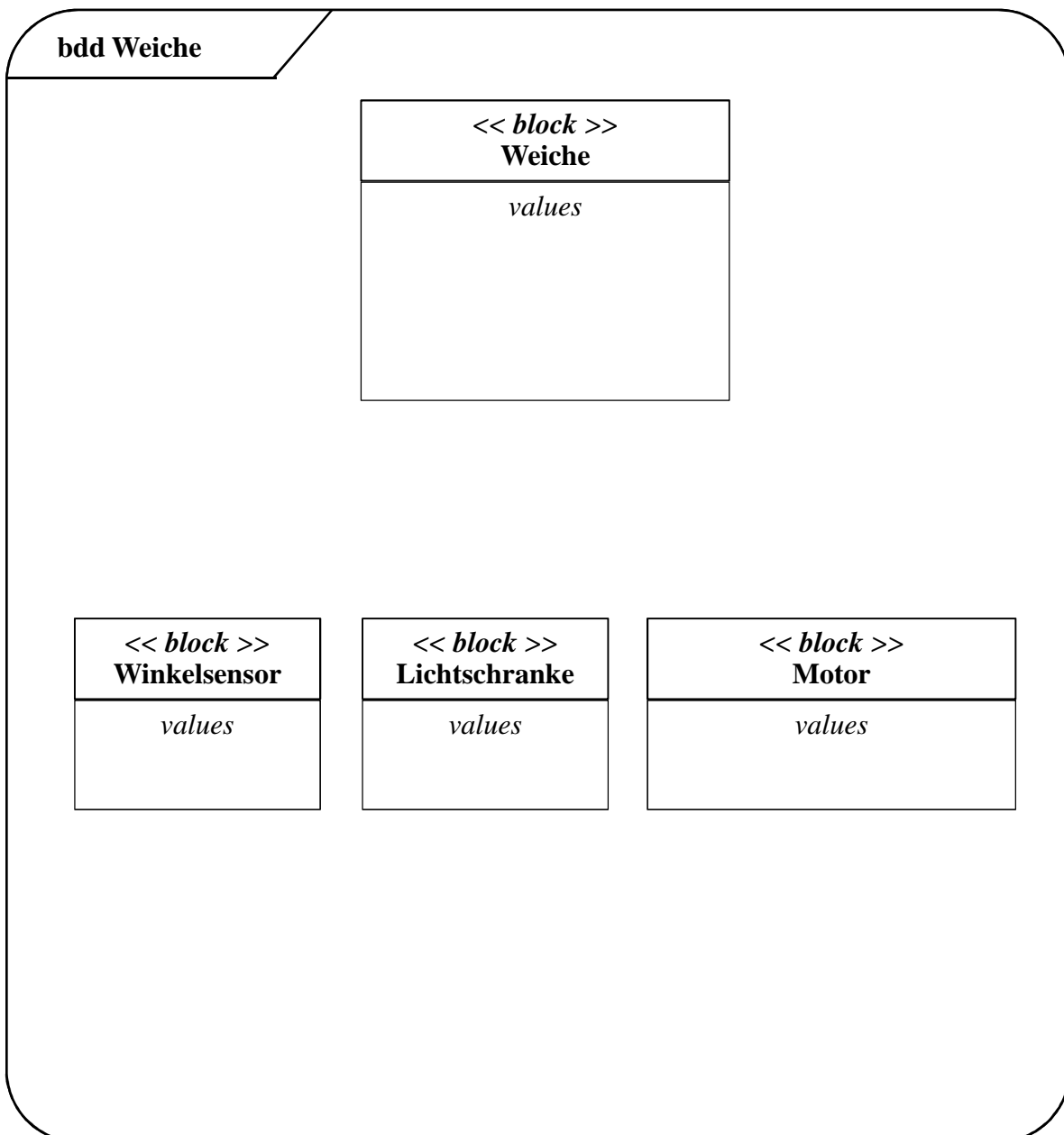


Vorname, Name

Matrikelnummer

Punkte

- b) Gegeben sei eine Weiche eines Transportsystems mit einem Motor sowie einem Sensor zur Erkennung der aktuellen Winkelstellung. Die aktuelle Winkelstellung der Weiche soll durch ein Attribut mit einem Fließkommazahlen-Datentyp modelliert werden. Der Winkelsensor und die Lichtschranke besitzen ein boolesches Signal, der Motor einen ganzzahligen Stellwert für die Geschwindigkeit. Fügen Sie diese Eigenschaften in das Blockdefinitionsdiagramm ein. Vervollständigen Sie darüber hinaus das unten stehende SysML Blockdefinitionsdiagramm durch die entsprechenden Beziehungen der Weiche zu den Untermodulen „Motor“ und „Winkelsensor“. Winkelsensor und Motor sollen dabei existenzabhängige Teile der Weiche sein. Des Weiteren besitzt die Weiche eine Referenz auf eine Lichtschranke. Geben Sie für die jeweiligen Beziehungen die Rollennamen (Instanznamen) sowie die Kardinalitäten (Multiplizitäten) an.





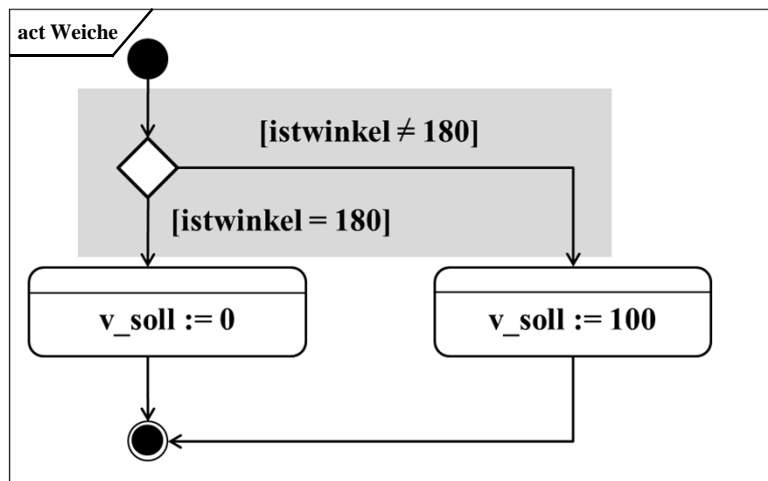
Vorname, Name

Matrikelnummer

c) Beantworten Sie die folgenden Fragen in dem Lösungskasten unten:

- Welches Kontrollfluss-Konstrukt ist in dem grau hinterlegten Kasten in dem folgenden Aktivitätsdiagramm abgebildet?
- Welches Mittel bietet hierzu die Programmiersprache C zur Implementierung an?
- Zeigen Sie die Syntax der Implementierung mit den Werten aus dem Beispiel und beantworten Sie die Frage nach der Notwendigkeit der geschweiften Klammern.

Punkte



Abgebildet ist: \_\_\_\_\_

Wird in C umgesetzt durch: \_\_\_\_\_

Implementierung:

```
int istwinkel;
int v_soll;
```

```

{
  _____
}
{
  _____
}

```

Sind in diesem Beispiel die geschweiften Klammern zwingend? Falls nicht, warum?

\_\_\_\_\_





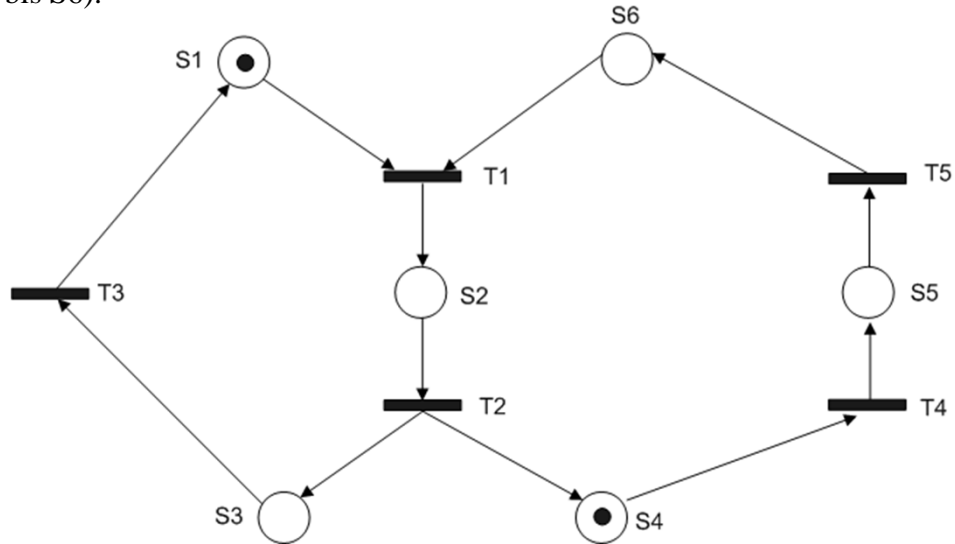
Vorname, Name

Matrikelnummer

Das unten abgebildete Petrinetz ist die Grundlage für die Aufgabe d).  
 Das Petrinetz ist mit seinen Markierungen im Startzustand abgebildet. Es gilt:  $K=1$  für alle Stellen (S1 bis S6).

Punkte

$$\underline{k} = \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix}$$



- d) Zeichnen Sie den Erreichbarkeitsgraphen des Petrinetzes und benennen sie die Transitionen.  
*Hinweis: Fehlerhafte Transitionen führen zu Punktabzügen.*

**M0 ( 1 0 0 1 0 0 )**



Vorname, Name

Matrikelnummer

**Aufgabe BS: Betriebssysteme**

*Aufgabe BS:  
60Punkte*

Punkte

- a) Auf einem Einprozessorsystem sollen die nachfolgend angegebenen Tasks ablaufen. Beantworten Sie hierzu die folgenden Fragen.

*Hinweis: Die Tabelle enthält alle wichtigen Angaben zur Planung der Tasks A,B,C. Die Ausführungsdauer ist eine relative Zeitangabe, wohingegen Bereit und Deadline als eine absolute Zeitangabe zu verstehen sind. Bei der Prioritätsangabe besitzt der Task mit dem niedrigsten numerischen Wert die höchste Priorität. Geben Sie, wenn gefragt, die Reihenfolge der ablaufenden Tasks an. z.B. ABCABB.*

Task	Ausführungsdauer	Bereit	Deadline	Priorität
A	3s	4s	8s	3
B	5s	0s	9s	1
C	6s	5s	13s	2

- 1) Geben Sie die Reihenfolge der Tasks und den Zeitpunkt an, an dem die Ausführung des letzten Tasks zum ersten Mal startet. Als Schedulingverfahren ist das FIFO-Verfahren einzusetzen.
- 2) Würde der Task B bei einem Least-Laxity Scheduling (LL) von Task A zum Zeitpunkt  $t = 4s$  unterbrochen werden? Begründen Sie Ihre Antwort stichpunktartig.
- 3) Wird bei einem Earliest Deadline First (EDF) Scheduling die Bedingung der Rechtzeitigkeit aller Tasks A,B,C erfüllt? Begründen Sie Ihre Antwort stichpunktartig.

1)

2)

3)



Vorname, Name

Matrikelnummer

Punkte

- b) Gegeben ist die Anordnung von Semaphor-Operationen am *Anfang und am Ende der Tasks A, B, C*. Ermitteln Sie für die Fälle I, II, III, *ob und in welcher Reihenfolge* diese Tasks bei der angegebenen Initialisierung der Semaphor-Variablen ablaufen. Geben Sie zusätzlich an, ob es sich bei der Taskreihenfolge um eine Wiederholungsreihenfolge handelt, oder ob ein Deadlock entstanden ist.

*Hinweis: Sind mehrere Tasks ablauffähig gelten folgende Prioritätslevel  $A = 1$ ;  $B = 2$ ;  $C = 3$ , wobei 1 die höchste Priorität ist. Geben Sie die Reihenfolge der ablaufenden Tasks in folgender Schreibweise an ABCABB an.  $P(S_i)$  senkt  $S_i$  um 1,  $V(S_i)$  erhöht  $S_i$  um 1. Pro richtigem Prozess wird 1 Punkt gegeben, sobald in einer Prozessreihe ein Fehler vorkommt, werden auf nachfolgende Prozesse keine Punkte mehr vergeben.*

Task	A	B	C	Fall	S1	S2	S3
	P(S3)	P(S2)	P(S1)	<b>I</b>	3	0	2
	P(S3)	P(S1)	P(S3)	<b>II</b>	2	2	0
	P(S3)			<b>III</b>	2	2	2
	...	...	...				
	V(S1)						
	V(S1)	V(S3)	V(S2)				
V(S1)	V(S3)	V(S2)					

**Fall I:** \_\_\_\_\_

**Fall II:** \_\_\_\_\_

**Fall III:** \_\_\_\_\_



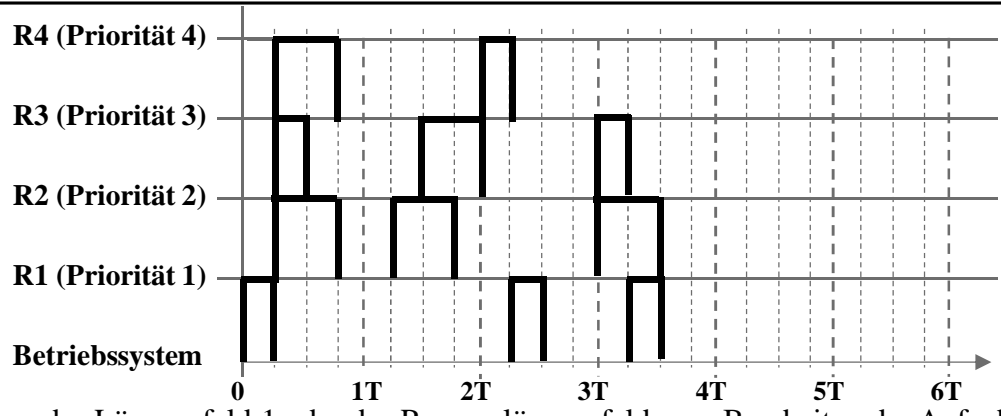
Vorname, Name

Matrikelnummer

Punkte

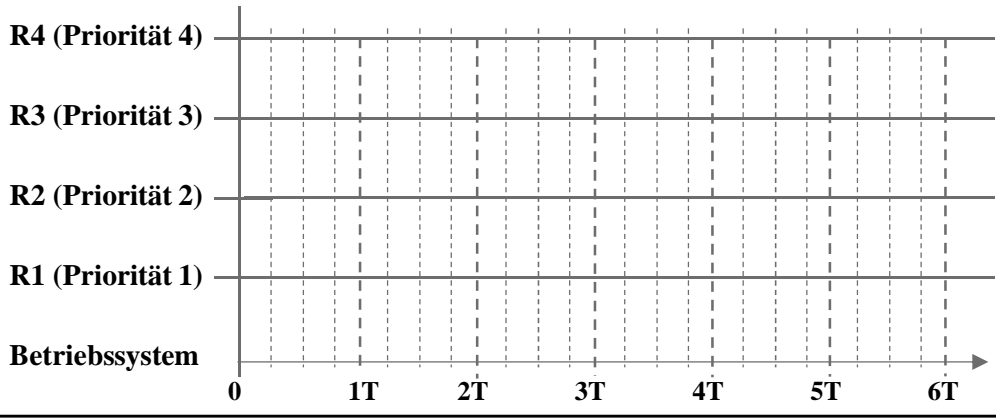
c) Stellen Sie das Ist-Systemverhalten der Programmierart *asynchron-preemptiv* in dem angegebenen Diagramm dar.

*Hinweis: Priorität 1 ist die höchste Priorität (BS > R1 > R2 > R3 > R4).*

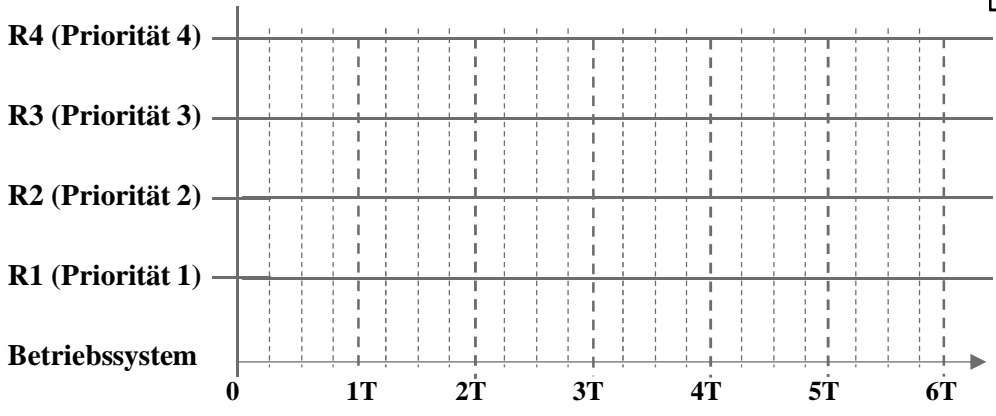


Es kann das Lösungsfeld 1 oder das Reservelösungsfeld zum Bearbeiten der Aufgabe genutzt werden. Sollten Sie das Reservelösungsfeld verwenden müssen, markieren Sie dies durch ein „X“ im Kästchen des Reservelösungsfelds. Das Lösungsfeld 1 wird nicht gewertet, sobald ein „X“ im Kästchen des Reservelösungsfelds angegeben ist!

**Asynchron-preemptive Programmierung: (Lösungsfeld 1)**



**Asynchron-preemptive Programmierung: Reservelösungsfeld werten**





Vorname, Name

Matrikelnummer

Punkte

- d) Für diese Aufgabe ist ein 2-dimensionales Array gegeben (`numberspace[2][10]`). Dieses beinhaltet – wie in der Abbildung unten zu sehen ist – die Zahlen von 1 bis 20. Das Array wird anschließend mit dem folgenden Algorithmus bearbeitet.

```

int i, j;
int currentReference = 3;
int nextReference = 3;
int currentValue;

while (1)
{
    for (i = 0; i < 2; i++)
    {
        for (j = 0; j < 10; j++)
        {
            currentValue = numberspace[i][j];

            if (currentValue == 1)
                numberspace[i][j] = NULL;

            else if (currentValue > currentReference)
            {
                if (currentValue % currentReference == 0)
                    numberspace[i][j] = NULL;

                else if (nextReference == currentReference)
                    nextReference = currentValue;
            }
        }
    }

    if (currentReference == nextReference)
        break;

    currentReference = nextReference;
}

```

Streichen Sie in der folgenden Abbildung alle Zahlen, welche durch diesen Algorithmus entfernt werden (auf NULL gesetzt).

*Hinweis: Pro falsch gestrichener Zahl gibt es einen Punkt Abzug. Es gibt keine negativen Gesamtpunkte.*

i	j →										
↓		1	2	3	4	5	6	7	8	9	10
		11	12	13	14	15	16	17	18	19	20



Vorname, Name

Matrikelnummer

Punkte

- e) Das unten angegebene Programm soll die Anzahl der Prozesse in der global definierten Variable `szProcessUIDs` gespeicherten Zeichenkette ermitteln und in der lokalen Variable `iTotalNumber` speichern. Die Prozess-UIDs sind dabei durch genau ein Symbol getrennt, welches entweder ein Leerzeichen (ASCII 32) oder ein Komma (ASCII 44) oder ein Punkt (ASCII 46) ist. Eine beispielhafte Zeichenkette mit 5 Prozess-UIDs wäre folgende:

```
char szProcessUIDs[] = "15,23 7 39.55";
```

Markieren und beschreiben Sie stichpunktartig die Fehler in folgender `main()`-Funktion eines C-Programms. Erläutern Sie kurz wofür die Zeilen 10 und 11 benötigt werden.

```
Zeilen
-nr.  :
      00: void main()
      01: {
      02: int iTotalNumber= 0;
      03: int i = 0
      04: while (szProcessUIDs[i] = '\0')
      05: {
      06:     if((szProcessUIDs[i] != 32)&(szProcessUIDs[i]== 44))
      07:         iTotalNumber++;
      08:     i++;
      09: }
      10: if(i)
      11:     iTotalNumber++;
      12: return 0;
      13: }
```

Zeile \_\_\_:

Zeile \_\_\_:

Zeile \_\_\_:

Zeile \_\_\_:

Zeile \_\_\_:

Zeile 10/11:



Vorname, Name

Matrikelnummer

**Aufgabe RK: Rechnerkommunikation**

*Aufgabe RK:  
55 Punkte*

Punkte

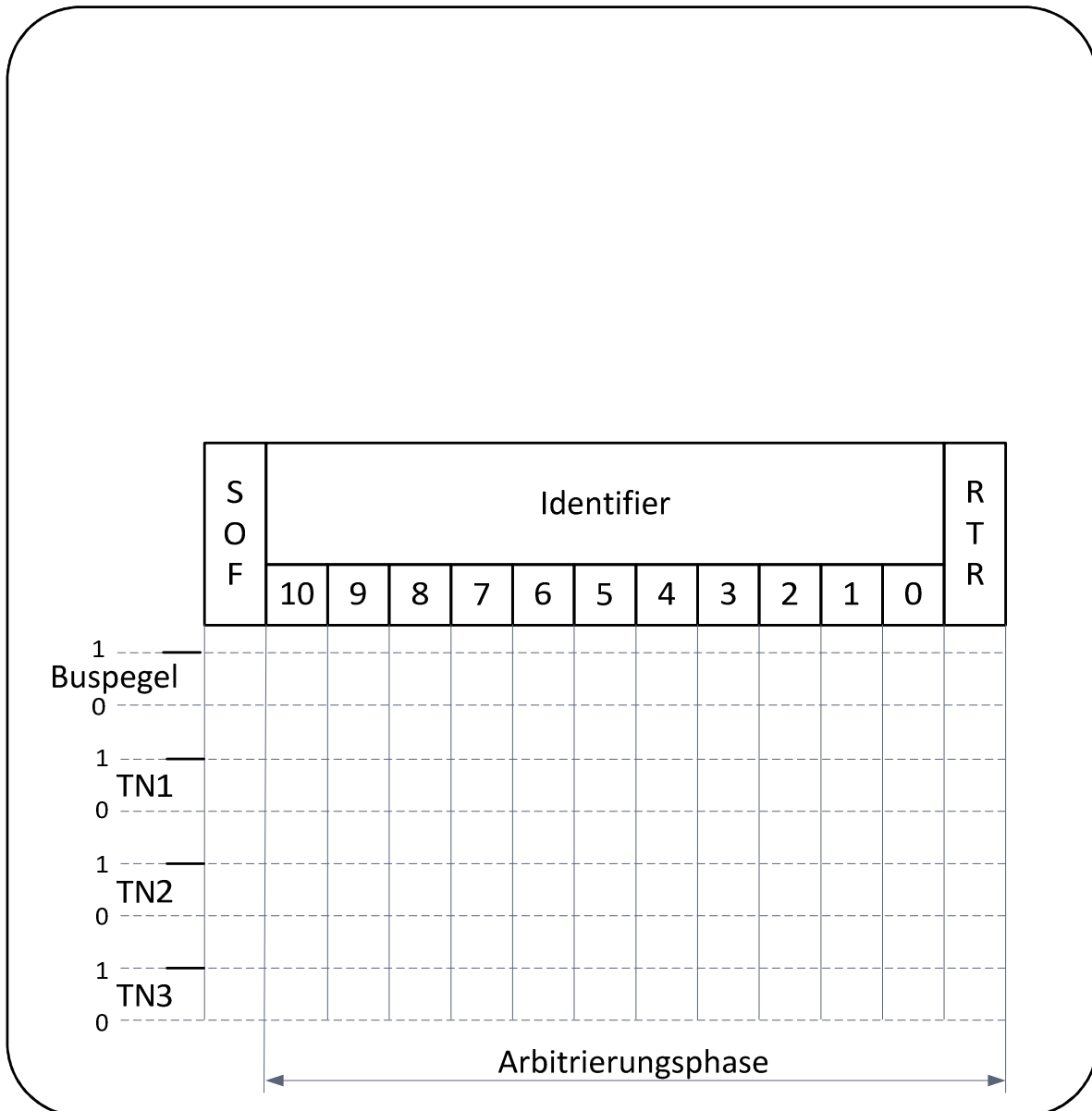
- a) Drei Teilnehmer sind an einem Bus mit CSMA/CA Verfahren angeschlossen (z.B. CAN) und wollen zum gleichen Zeitpunkt senden. In der Arbitrierungsphase senden alle Teilnehmer die folgend angegebenen Identifier:

Teilnehmer 1: 34F (hex)

Teilnehmer 2: 24E (hex)

Teilnehmer 3: 267 (hex)

Stellen Sie im folgenden Diagramm den Arbitrierungsvorgang dar und geben Sie an, welcher Teilnehmer zu welchem Takt aus dem Arbitrierungsprozess ausscheidet. Stellen Sie weiterhin den resultierenden Buspegel im Diagramm dar. Null ist dominant.





\_\_\_\_\_  
Vorname, Name

\_\_\_\_\_  
Matrikelnummer

Punkte

b) Nennen Sie zwei Vorteile der Schnellanschlusstechnik.

c) Nennen Sie zu jedem Buszugriffsverfahren ein entsprechendes Beispiel.  
Ein Beispiel kann (falls zutreffend) mehrmals verwendet werden.

Buszugriffsverfahren	Beispiel
zentral gesteuert	
dezentral gesteuert	
kollisionsfrei ungesteuert	
nicht kollisionsfrei ungesteuert	





\_\_\_\_\_  
Vorname, Name

\_\_\_\_\_  
Matrikelnummer

Punkte

- d) Benennen Sie die zwei Maßnahmen der Fehlersicherung und geben Sie jeweils mindestens einen Methodenbeispiel. Erklären Sie stichpunktartig die Wirkungsweise jeweiliger Maßnahme.



\_\_\_\_\_  
Vorname, Name

\_\_\_\_\_  
Matrikelnummer

Punkte

- e) Eine Wetterstation speichert aktuelle Messwerte in einem Array mit dem Namen Messwerte. Das Array selbst ist wie folgt aufgebaut:

Messgröße	Typ	Mögliche Werte
Temperatur	Integer (32Bit)	ganzzahlig 0-100
Luftdruck	Integer (32Bit)	ganzzahlig 500-1100
Regenstärke	Char	0 oder 1

Temperatur und Luftdruck werden nur ganzzahlig und nicht negativ gemessen. Der Rechner der Wetterstation legt die Werte in der oben angegebenen Reihenfolge direkt aufeinanderfolgend im „Little Endian“ Format.

**Allgemeine Fragen:**

Die Zahl 513 wird in einer 32 Bit Integer Variablen gespeichert (Little Endian!). Tragen Sie die Werte der einzelnen Bytes in die Kästchen ein:

Adresse:	100	101	102	103
Wert:				

Warum ist in diesem Beispiel der Datentyp Integer für die Temperatur eine schlechte Wahl: \_\_\_\_\_

Welcher Datentyp wäre dafür besser geeignet: \_\_\_\_\_

**Zur Implementierung in C:**

Definieren Sie ein Array vom Typ char mit dem Namen „Messwerte“, das alle drei Messwerte hintereinander aufnehmen kann:  
\_\_\_\_\_

Definieren Sie einen Zeiger auf Integer mit dem Namen pLuftdruck, initialisiert mit dem Wert 0:  
\_\_\_\_\_

Biegen Sie den Zeiger auf den Luftdruck im Array:  
\_\_\_\_\_

Überschreiben Sie den Luftdruck im Array unter Verwendung des Zeigers mit dem Wert 1234: \_\_\_\_\_

Überschreiben Sie den Wert des Regensensors im Array ohne Verwendung einer zusätzlichen Zeigervariablen mit dem Wert 1:  
\_\_\_\_\_



Vorname, Name

Matrikelnummer

Punkte

f) Überführen Sie die main-Funktion in ein Nassi-Shneiderman-Diagramm.

```
int main(void)
{
    unsigned int iZaehler = 0, iMaxNetzwerkkarten = 0, iAnzahlRechner = 0, iIndex = 0;
    printf("\nGeben Sie die Anzahl der Computer ein: ");
    scanf("%d", &iAnzahlRechner);
    iIndex = 1;
    do
    {
        printf("\nAnzahl der Netzwerkkarten des %d-ten Computers eingeben: ", iIndex);
        scanf("%d", &iZaehler);
        if(iZaehler > iMaxNetzwerkkarten)
            iMaxNetzwerkkarten = iZaehler;
        iIndex++;
    } while (iIndex <= iAnzahlRechner);
    printf("\nDie groesste Anzahl an Netzwerkkarten ist %d!\n", iMaxNetzwerkkarten);
    return 0;
}
```

\_\_\_\_\_  
Vorname, Name\_\_\_\_\_  
Matrikelnummer**Aufgabe DB: Datenbanken****Aufgabe DB:  
35 Punkte**

Punkte

- a) Zeichnen Sie ein Entity-Relationship-Diagramm. Ergänzen Sie sinnvolle Primärschlüssel. Geben Sie die entsprechenden Kardinalitäten an.
- Eine deutsche Boulevardzeitung möchte ihre Anrufbeantworternachrichten archivieren. Aufgrund der Vielzahl möchte sie dafür eine Datenbank aufsetzen. Die aufgesprochenen *Nachrichten* kommen von einem *Anrufer* mit Name sowie Beruf (z.B. Politiker, Polizist ...) und haben eine bestimmte Länge mit Inhalt. Ein *Anrufer* kann beliebig viele *Nachrichten* hinterlegen. Eine *Nachricht* kann beliebig viele *Adressaten* haben und dieser kann in mehreren *Nachrichten* vorkommen. Die *Adressaten* sollen mit Namen und Adresse in der Datenbank hinterlegt werden. Um später nach Drohanrufen sortieren zu können soll einer *Nachricht* außerdem eine Kritikalität zugeordnet werden können. Die Kritikalität kann sich während eines Gesprächs ändern und soll mit einem Zeitstempel gespeichert werden.



Vorname, Name

Matrikelnummer

Punkte

Für alle weiteren Teilaufgaben ist folgender unvollständiger Datenbankausschnitt gegeben. Der folgende Datenbankausschnitt steht in keinem Zusammenhang mit der Teilaufgabe a).

Absender		
Name	Beruf	<u>AID</u>
Peter Hudson	Politiker	1
Karl Lewis	Staatsanwalt	2
Charlie Runkle	Schauspieler	3

Gesprochen	
<u>AID</u>	ADID
1	1
2	3
3	2

Bezug	
<u>NID</u>	ADID
1	1
2	3
3	2

Nachricht			
<u>NID</u>	Inhalt	Laenge	Kritikalitaet
1	Text	5.2	5
2	Text	10.0	8
3	Text	12.3	10

Adressat	
ADID	Name
1	Kai D.
3	Friede S.
2	Herbert G.

b) Legen Sie mit einer SQL-Anweisung die Tabelle *Nachricht* an.



\_\_\_\_\_  
Vorname, Name

\_\_\_\_\_  
Matrikelnummer

Punkte

c) Befüllen Sie die Tabelle *Nachricht* mit einer SQL-Anweisung mit folgenden Daten:

- Laenge: *12,30*
- Inhalt: *Text*
- NID: *3*
- Kritikalitaet: *10*

d) Geben Sie eine SQL-Anweisung an, mit der der *Name* des *Absenders* angezeigt wird, der eine *Nachricht* mit der *Kritikalitaet* 10 hinterlassen hat.