

Institut für Luftfahrtsysteme (ILS)  
Universität Stuttgart  
&  
Institut für Flugregelung (IFR)  
Universität Stuttgart

Softwarewerkzeuge und Softwaretechnik  
(Software Tools and Software Engineering)

Ort: V 07.02, V 57.03, V 47.01  
Datum: Donnerstag, 29. September 2016  
Bearbeitungszeit: 120 Minuten  
Anzahl Seiten: 35 abzgl. 1 Seite (Deckblatt)

Diese Angaben sind sorgfältig und in Druckbuchstaben auszufüllen!

Matrikelnummer:

Name:

Vorname:

Studiengang: ☐ aer ☐ geod ☐ kyb

**Hinweise und Bewertung:**

Die Prüfung besteht aus Single Choice Fragen, Programmier- und MATLAB-Aufgaben. Pro Single Choice Frage ist immer genau eine Antwort richtig und wird mit einem Punkt gewertet. Falsche, keine oder mehrere Antworten werden mit null Punkten gewertet. Bei den Programmieraufgaben und MATLAB-Aufgaben ergeben syntaktische und semantische Fehler Abzug von der maximal erreichbaren Punktezahl der jeweiligen Aufgabe.

- Achten Sie immer darauf die korrekte Syntax in Ihren Lösungen zu verwenden (z.B. bei mathematischen Operationen).
- Achten Sie bei der Verwendung von Klammern auf eine klare Darstellung (runde Klammern, eckige Klammern,...).
- Füllen Sie jeweils die Lösungen in die dafür vorgesehenen grauen Kästen.

**Zugelassene Hilfsmittel:**

Es sind keine Hilfsmittel zugelassen.

Korrekturhilfe (nur bei der Korrektur der Prüfung ausfüllen):

SC	T1	T2	T3	T4	P1	P2	P3	P4	P5	M	SUM	ÜB



**Aufgabe 1:** Gegeben ist folgendes Programm in C. Was wird ausgegeben?

```
void vergleichendeOperatoren(void) {  
    int a = 10 > 0; true = 1  
    int b = 10 < 0; false = 0  
    int c = 0 != 1; true = 1  
    int d = 10 <= 10; true = 1  
    int e = 10 >= 10; true = 1  
    int f = 10 == 10; true = 1  
    printf("%d %d %d %d %d %d\n", a, b, c, d, e, f);  
}  
int main(void) {  
    vergleichendeOperatoren();  
    return EXIT_SUCCESS;  
}
```

- ☒ 1 0 1 1 1 1  
☐ 0 1 0 0 0 0  
☐ 1 0 1 0 0 1

**Aufgabe 2:** Gegeben ist folgendes Programm in C. Was wird ausgegeben?

```
void logischeOperatoren(void) {  
    int a = 10 && 5; true && true = true = 1  
    int b = 10 || 5; true || true = true = 1  
    int c = !10; false = 0  
    printf("%d %d %d\n", a, b, c);  
}  
int main(void) {  
    logischeOperatoren ();  
    return EXIT_SUCCESS;  
}
```

- ☒ 1 1 0  
☐ 0 0 1  
☐ 0 1 0

**Aufgabe 3:** Welches Header-File muss eingebunden werden, damit die Funktionsdeklaration der Funktion printf und scanf verfügbar sind?

- ☐ stdlib.h  
☒ stdio.h  
☐ stddef.h



**Aufgabe 4:** Gegeben ist folgendes Programm in C. Was wird ausgegeben?

```
void bitwiseOperatoren(void) {
    int a = 10 & 5; 10 = 0b00001010; 5 = 0b00000101; res = 0
    int b = 10 | 5; 0b00001010;
    int c = ~10+1; 0b00000101;
    int d = 10>>1; -----
    int e = 10<<1; 0b00001111; = 15

    printf("%d %d %d %d %d \n", a, b, c, d, e);
}

int main(void) {
    bitwiseOperatoren();
    return EXIT_SUCCESS;
}
```

- ☐ 0 16 -10 6 20  
☒ 0 15 -10 5 20  
☐ 0 1 -11 5 20

**Aufgabe 5:** Ein Programm soll ausgeben ob es sich bei der eingegebenen Jahreszahl um ein Schaltjahr handelt. Ein Schaltjahr liegt vor, wenn die aktuelle Jahreszahl durch vier (ohne teilbar ist und kein volles Jahrhundertjahr ist oder durch 400 teilbar ist. Ergänzen Sie das rest) Programm an der gekennzeichneten Stelle!

```
int main(void) {
    int jahr;
    scanf("%d", &jahr);
    >>> Hier fehlt was, bitte ergaenzen
    return EXIT_SUCCESS;
}

☐ printf("Schaltjahr %s", ((jahr/4==0)&&(jahr/100!=0)) || (jahr/400==0)?"Ja":"Nein");
☒ printf("Schaltjahr %s", ((jahr%4==0)&&(jahr%100!=0)) |!(jahr%400==0)?"Ja":"Nein");
☐ printf("Schaltjahr %s", ((jahr%4==0) || (jahr%100!=0)) && (jahr%400==0)?"Ja":"Nein");
```

**Aufgabe 6:** Gegeben ist folgendes Programm in C. Welche Fehlerart liegt hier vor?

```
int main(void)
{
    int Airbus = 5.0;
    return EXIT_SUCCESS;
}
```

- ☒ Syntaktischer  
☐ Kein Fehler  
☒ Semantischer



**Aufgabe 7:** Gegeben ist folgendes Programm in C. Was wird ausgegeben?

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void) {
    char wort[10];
    char string1[]="soft\0ware";
    char string3[]="bild";
    char string4[10];
    char string5[10];
    char string6[30]="";
    strcpy(string4,"schirm");
    strcpy(string5,"steuerung");
    strcat(string6,string1);
    strcat(string6,string3);
    strcat(string6,string4);
    printf("%s",string6);
    printf("%s",string5);
    return EXIT_SUCCESS;
}

```

- ☐ softbildschirmsteuerung  
☒ softwarebildschirmsteuerung  
☐ bildschirmsoftsteuerung

**Aufgabe 8:** Folgendes Programm gibt eine Datei mit Zeilennummern auf der Konsole aus. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```

#include <stdio.h>
#include <stdlib.h>
int main(int argc,char * argv[],char * envp[] ) {
    FILE *fileId;
    int zeile=0;
    int zeichen;
    fileId=fopen(argv[1],"r");
    if(fileId==NULL){
        printf("Kann die Datei %s nicht öffnen\n",argv[1]);
        exit(EXIT_FAILURE);
    }
    printf("%4d:",zeile);
    zeichen=fgetc(fileId);
    >>> Hier fehlt was, bitte ergaenzen
    printf("%c", (char) zeichen);
    if(zeichen == '\n'){
        zeile++;
        printf("%4d:",zeile);
    }
    zeichen=fgetc(fileId);
}
fclose(fileId);
return EXIT_SUCCESS;

```

- ☐ while(zeichen == EOF){  
☒ while(!(zeichen == EOF)){  
☐ while(zeichen == EOF){



**Aufgabe 9:** Auf ein Arrayelement wird mit welchem Operator in C zugegriffen?

- ☐ .  
☐ ->  
☒ []
- ```
int array[] = {}
element = array[0]
```

**Aufgabe 10:** Gegeben ist folgendes Programm in C. Was wird auf der Konsole ausgegeben?

```
#include <stdio.h>
#include <stdlib.h>
int x=5;
int zahl1(void) {
    int x =20;
    return x;
}
int zahl2 (void) {
    int x=10;
    return x;
}
int zahl3 (void) {
    return zahl2();
}
int main(void) {
    printf("%i,%i,%i,%i",x,zahl1(),zahl2()+zahl3(),zahl2()+zahl1());
    return EXIT_SUCCESS;
}
```

- ☒ 5,20,20,30  
☐ 5,20,10,30  
☐ 5,10,10,30

**Aufgabe 11:** Gegeben ist folgendes Programm in C. Was wird auf der Konsole ausgegeben?

```
int main(void) {
    char str1[]={'S','\0','o','f','t','\0','w','a','r','e','\0'};
    printf("%s",str1);
    return EXIT_SUCCESS;
}
```

- ☐ Software  
☒ S  
☐ Soft

**Aufgabe 12:** Rechnen Sie die Dezimalzahl 51 ins Dualsystem um.

- |                                     |                     |                      |
|-------------------------------------|---------------------|----------------------|
| <input type="checkbox"/>            | 111001 <sub>2</sub> | 00000000             |
|                                     |                     | 128 64 32 16 8 4 2 1 |
| <input checked="" type="checkbox"/> | 110011 <sub>2</sub> | 0 0 1 1 0 0 1 1      |
| <input type="checkbox"/>            | 100111 <sub>2</sub> |                      |



**Aufgabe 13:** Folgendes Programm rechnet das Quadrat einer gegebenen Zahl aus. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```
void square ( long *v) /* Funktionsdefinition*/
{
  >>> Hier fehlt was, bitte ergaenzen
}
int main(void) {
  int i=3;
  square(&i);
  return EXIT_SUCCESS;
}
```

- ☐ `v *= *v; return v;`  
☒ `*v *= *v;`  
☐ `*v = v*v;`

**Aufgabe 14:** Gegeben ist folgendes Programm in C. Was wird auf der Konsole ausgegeben?

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(void) {
  char satz[200]= "Softwarewerkzeuge fuer Ingenieure";
  int anzahl=0;
  int laenge;
  int i;
  laenge=strlen(satz); laenge = 33
  for(i=0;i < laenge;i++){
    switch(satz[i]){
      case 'a':
      case 'e':
      case 'i':
      case 'o':
        anzahl++;break;
      default:
        break;
    }
  }
  printf("%d",anzahl);
  return EXIT_SUCCESS;
}
```

- ☐ 12  
☒ 11  
☐ 1

**Aufgabe 15:** Rechnen Sie die Zahl  $10100111110000111_2$  im Binärsystem in das Hexadezimalsystem um.

$$1. 10 = A$$

$$8 = 0x08$$

- ☐ B757<sub>16</sub>  
☒ A787<sub>16</sub>  
☐ A747<sub>16</sub>



**Aufgabe 16:** Die Funktion in C erstellt einen neuen Knoten und soll diesen mit 0 initialisieren. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```

Node * Node_New(void) {
    Node * newNode;
    >>> Hier fehlt was, bitte ergaenzen
    return newNode;
}
☐ newNode=(Node*) malloc(sizeof(Node));memset(&newNode,0,sizeof(Node));
☒ newNode=(Node*) malloc(sizeof(Node));memset(newNode,0,sizeof(Node));
☐ newNode=(Node*) malloc(sizeof(Node));memset(*newNode,0,sizeof(Node));

```

**Aufgabe 17:** Das folgende Programm führt eine Passwort-Abfrage durch und prüft, ob dieses dem hinterlegten Passwort entspricht. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

int main(void) {
    char geheim[10];
    char passwort[30];
    int versuch=3;
    geheim[3]= 'p';
    geheim[2]=109;
    geheim[7]=114;
    geheim[5]=116;
    geheim[0]=67;
    geheim[9]='x';
    geheim[1]=111;
    geheim[8]='\0';
    geheim[4]=geheim[1]+6;
    geheim[6]=101;
    printf("\n Sie haben 3 Versuche\n");
    do{
        printf("\nBitte Passwort eingeben:");
        gets(passwort);
        >>> Hier fehlt was, bitte ergaenzen
        printf("\n Willkommen zurück!\n");
        versuch=0;
    }else{
        printf("\nFehlerhaftes Passwort");
        versuch--;
        printf("\nSie haben noch %d Versuche",versuch);
    }
    }while(versuch >0);
    return EXIT_SUCCESS;
}

☐ if(geheim == passwort){
☒ if(strcmp(geheim,passwort)==0){
☐ if(strcmp(geheim,passwort) != 0){

```



**Aufgabe 18:** Wie lautet die Merkregel für Pointer in C?

- ☐ Zeigt der Pointer p auf die Variable x, so kann p-> im Programm überall dort stehen, wo x gebraucht wird.
- ☐ Zeigt der Pointer p auf die Variable x, so kann &p im Programm überall dort stehen, wo x gebraucht wird.
- ☒ Zeigt der Pointer p auf die Variable x, so kann \*p im Programm überall dort stehen, wo x gebraucht wird.

XOR

**Aufgabe 19:** Die beiden Binärzahlen sollen miteinander **bitweise** ENTWEDER ODER verknüpft werden. Was kommt raus?

$$11101010_2 \text{ XOR } 01010101_2$$

11101010

01010101

\_\_\_\_\_

10111111

- |                                     |              |
|-------------------------------------|--------------|
| <input type="checkbox"/>            | $11101111_2$ |
| <input type="checkbox"/>            | $01000000_2$ |
| <input checked="" type="checkbox"/> | $10111111_2$ |

**Aufgabe 20:** Was bewirkt die Schleife im folgenden Programm in C?

```
int main(void)
{
    int i=0;

    for(;;i++);
    return(0);
}
```

- ☐ Eine nicht gültige Anweisung in C.
- ☐ Eine Schleife die keine Auswirkung hat.
- ☒ Eine Endlosschleife.

**Aufgabe 21:** Folgende komplexe Datenstruktur ist gegeben. Welche Definition für ein Array von 100 Elementen des Datentyps Flugzeug ist **nicht** richtig?

```
typedef struct flugzeug
{
    char hersteller [10];
    char name [10];
    float preis;
}Flugzeug;
```

- ☐ struct flugzeug flugzeuge[100];
- ☒ struct Flugzeug flugzeuge[100];
- ☐ Flugzeug flugzeuge[100];



**Aufgabe 22:** Gegeben ist folgendes Programm in C. Was gibt das Programm auf der Konsole aus?

```
#include <stdio.h>
#include <stdlib.h>
int main ( void ){
    int x ;
    for(x=1; x <= 10; x++){
        if(x%4 == 0){
            /* do nothing */
        }else{
            printf("%5d",x);
        }
    }
    printf("\n");
    return EXIT_SUCCESS;
}
```

- ☐ 4 8
- ☒ 1 2 3 5 6 7 9 10
- ☐ 1 2 3 5 6 7 8 9 10

**Aufgabe 23:** Rechnen Sie die Zahl  $A320_{16}$  im Hexadezimalsystem in das Binärsystem um.

- ☐  $A = 10 = 0b1010$
- ☒  $1010011100110000_2$   $3 = 3 = 0b0011$
- ☒  $1010001100100000_2$   $2 = 2 = 0b0010$
- ☐  $1010001100010000_2$   $0 = 0 = 0b0000$

**Aufgabe 24:** Aus einem C-Programm ist folgende Variablenvereinbarung bekannt:

```
int k=6, *j, *p, q[10];
```

Welche Aussage ist korrekt?

- ☐ Variablen k und q sind vom Typ Integer, p und j sind Pointer vom Typ Integer.
- ☐ Variable q und j sind vom Typ Integer, p und k sind Pointer vom Typ Integer.
- ☒ Variable k ist vom Typ Integer, q ist ein Integer-Array, p und j sind Pointer vom Typ Integer.

**Aufgabe 25:** Gegeben ist folgende Datenstruktur in C. Welche Aussage in Bezug auf die Datenstruktur ist richtig?

```
struct MenuItem{
    char titel[STR_LENGTH];
    void (* pf)(void);
};
```

- ☒ pf ist ein Zeiger auf eine Funktion mit dem Namen pf, der als Parameter nichts übergeben wird und die nichts zurückgibt.
- ☐ pf ist ein Zeiger auf eine Funktion mit dem Namen void, der als Parameter pf übergeben wird und die nichts zurückgibt.
- ☐ pf ist ein Zeiger auf eine Funktion mit dem Namen pf, der als Parameter ein Integer übergeben wird und die nichts zurückgibt.



**Aufgabe 26:** Das folgende C-Programm stellt einen einfachen Taschenrechner dar. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```
#include <stdio.h>
#include <stdlib.h>
int main(void) {
    float zahl1,zahl2;
    char operator = '0';
    int i;
    for (i=0;i < 5 || operator != 'q';i++){
        printf("Ein kleiner Taschenrechner:\n");
        printf("Geben Sie ein: Zahl operator Zahl <Return>\n");
        >>> Hier fehlt was, bitte ergaenzen
        if(operator == '+')
            printf("= %f", zahl1+zahl2);
        else if (operator == '-')
            printf("= %f",zahl1-zahl2);
        else if (operator == '*')
            printf("= %f", zahl1 * zahl2);
        else if( operator == '/')
            printf("= %f", zahl1/zahl2);
        else
            printf("Operator nicht bekannt");

        printf("\n\n");
    }
    return EXIT_SUCCESS;
}
☐ scanf("%f %c %f",zahl1,operator,zahl2);
☒ scanf("%f %c %f",&zahl1,&operator,&zahl2);
☐ scanf("%f %c %f",&zahl1,operator,&zahl2);
```

**Aufgabe 27:** Gegeben ist folgender C-Programm-Ausschnitt. Was wird ausgegeben?

```
#include <stdio.h>
#include <stdlib.h>
int main(void)
{
    for ( printf("%c",'A'); printf( "%c",'3'); printf( "%c",'8'))
        printf( "%c",'0');
}
☒ A308308308308308308308308308...
☐ A38080808080808080808080808...
☐ A380380380380380380380380380...
```

**Aufgabe 28:** Das folgende C-Programm erstellt eine Variable vom komplexen Datentyp person und soll den Inhalt der Variable mit Hilfe von printf ausgeben. Ergänzen Sie das Programm an der gekennzeichneten Stelle!

```
#include <stdio.h>
#include <string.h>

struct person{
    char name[20]
    int plz;
```



```
char ort[20];
};
```

```
void main(void) {
struct person leser = {"Otte", 5100, "Karlsbad"};
struct person *pleser;
pleser = &leser;
>>> Hier fehlt was, bitte ergaenzen
}
```

- ☒ printf("Name:%s,PLZ:%d,Ort:%s", leser.name, pleser->plz, leser.ort);
- ☐ printf("Name:%s,PLZ:%d,Ort:%s", leser->name, pleser->plz, leser->ort);
- ☐ printf("Name:%s,PLZ:%d,Ort:%s", pleser->name, pleser.plz, pleser->ort);

**Aufgabe 29:** Gegeben ist folgendes Programm in C. Übertragen Sie die Ausgabe des Programms in die grauen Felder der Tabelle! Start und die Laufvariable i sind in der Tabelle schon eingetragen. Tragen Sie für die einzelnen Programmschritte die Werte für x, y und z in die Tabelle ein. (4 Punkte)

```
#include <stdio.h>
#include <stdlib.h>
int i=1;
long errate(int x, int y) x = 5, y = 4
{
    long z=0;
    int i=1;
    printf("Start %d %d %d\n", x, y, z);    z = 4
    while(x) {                             x = 1
        if(x%2) z+=y;                       y = 32
        x/=2;
        y*=2;
        printf("%d %d %d %d\n", i++, x, y, z);
    }
    return z;
}
int main(void) {
    errate(5, 4);
    return EXIT_SUCCESS;
}
```

| Start | x | y  | z  |
|-------|---|----|----|
| 1     | 5 | 4  | 0  |
| 2     | 4 | 2  | 16 |
| 3     | 1 | 32 | 4  |



**Aufgabe 30:** Das folgende C-Programm stellt verschiedene Makros zur Verfügung. Mit dem main Programm sollen die Makro-Funktionen getestet werden. Tragen Sie in die grauen Felder der Tabelle die jeweiligen Ausgaben einer Ausführung ein! **(4 Punkte)**

```
#include <stdio.h>
#include <stdlib.h>

#define MAX(a,b) (a)>(b)?(a):(b)
#define MIN(a,b) (a)<(b)?(a):(b)
#define ABS(a) (a)<0?- (a):(a)
#define QUADRAT(a) a*a      20+20+1

int main(void) {
    int x=10;
    int y=20;

    printf("MAX %d\n",MAX(x,y));
    printf("ABS %d\n",ABS(-10));
    printf("MIN %d\n",MIN(x,y));
    printf("QUADRAT %d\n",QUADRAT(y));
    printf("QUADRAT %d\n",QUADRAT(y+1));
    printf("ABS %d\n",ABS(x-y));
    printf("MAX %d\n",MAX(x,y++));
    printf("y %d\n",y);
    return EXIT_SUCCESS;
}
```

| Ausgabe |                 |
|---------|-----------------|
| MAX     | 20              |
| ABS     | 10              |
| MIN     | 10              |
| QUADRAT | 400             |
| QUADRAT | 21 <sup>2</sup> |
| ABS     | 10              |
| MAX     | 20              |
| y       | 21              |



**Aufgabe 31:** Gegeben ist folgendes Programm in C. Übertragen Sie die Ausgabe des Programms in die grauen Felder der Tabelle! (5 Punkte)

```
#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 5

char a[ARRAY_SIZE]={21,5,1,11,14};

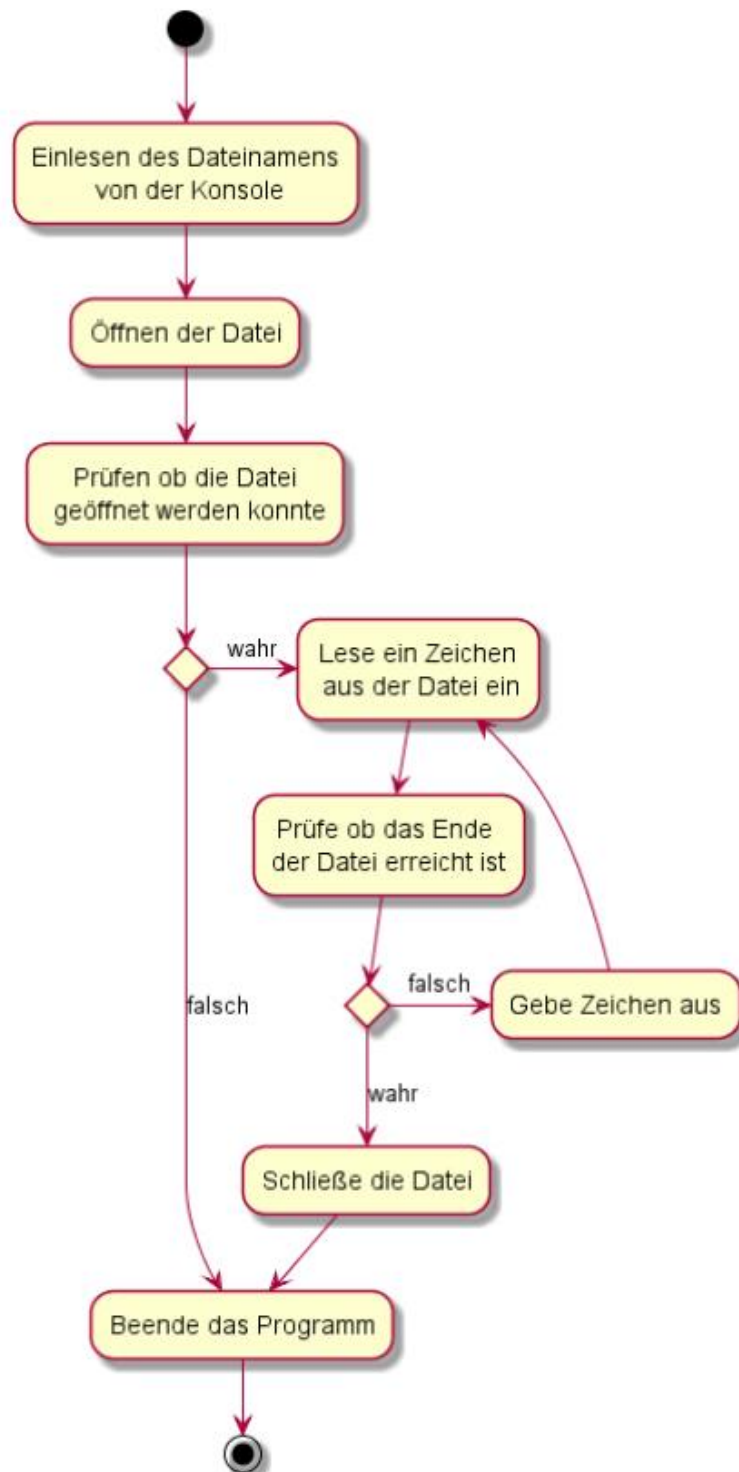
void sort(void)
{
    int i,j,min,t;
    for(i=0;i<ARRAY_SIZE;i++){
        printf("%d %d %d %d %d",a[0],a[1],a[2],a[3],a[4]);
        min = i;
        for(j=min;j<ARRAY_SIZE;j++)
            if(a[j] < a[min]) min=j;
        t=a[min];
        a[min]=a[i];
        a[i]=t;
    }
}

int main(void) {
    sort();
    return EXIT_SUCCESS;
}
```

|  |  |  |  |  |
|--|--|--|--|--|
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |



**Aufgabe 32:** Gegeben ist folgender Algorithmus für das zeichenweise Auslesen einer Datei in Form eines Activity Charts. Übertragen Sie den Algorithmus in ein C-Programm. Die Funktion `main` und die verwendeten Variablen sind vorgegeben. **(max. 10 Punkte)**  
**Folgende Funktionen sollen verwendet werden:** `gets`, `fgetc`, `fopen`, `fclose`, `exit`, `putchar`



```
#include <stdio.h>
#include <stdlib.h>

/* laenge des Dateinamen */
#define DATEINAME_LENGTH 20

int main(void)
{
    FILE * dateiPtr;
    char zeichen;
    char dateiname[DATEINAME_LENGTH];
```

```
return EXIT_SUCCESS;
}
```



## C-Programmieraufgaben

Die Programmieraufgaben behandeln eine Datenbank zur Verwaltung von Drohnen und deren Eigenschaften, wie im Folgenden dargestellt.



Abbildung 1: Beispielbild einer Quadcopterdrohne (<http://www.drohnen.de/4012/dji-inspire-1-test-vergleich-bewertung>)

Eine Drohne hat eine beliebige Anzahl von Eigenschaften. Die Liste der Eigenschaften wurde für die Prüfung begrenzt und stellt einen Ausschnitt dar. In der folgenden Tabelle sind die Eigenschaften mit den Werten einer „DJI Inspire 1 Quadcopter“ aufgelistet.

| Eigenschaft                  | DJI Inspire 1 Quadcopter |
|------------------------------|--------------------------|
| Model                        | T600                     |
| Hovering Accuracy Vertical   | 0.5m                     |
| Hovering Accuracy Horizontal | 2.5m                     |
| Max Ascent Speed             | 5 m/s                    |
| Max Descent Speed            | 4 m/s                    |
| Max Speed                    | 22 m/s                   |
| Max Flight Time              | 18 minutes               |
| Max Flight Altitude          | 4500 m                   |

Tabelle 1: Grunddatenstruktur

In den Aufgaben sind Teile der Umsetzung in der Programmiersprache C zu ergänzen oder komplette Teile in der Programmiersprache C neu zu programmieren. Für die Lösung der Aufgaben verwenden Sie die zur Verfügung gestellten Kästen. Die grau markierten Bereiche innerhalb der Kästen zeigen die auszufüllenden Bereiche an. Es soll nur ANSI-C99 verwendet werden. Die Drohrendatenbank verwaltet eine beliebige Anzahl von Drohnen in einer doppelt verketteten Liste. Das Programm hat zur vereinfachten Benutzung eine menügesteuerte Benutzerführung. In der Aufgabe wird davon ausgegangen, dass der Name einer Drohne keine Leerzeichen enthält, bzw. der Benutzer diese, wenn nötig, durch Bindestrich ersetzt. (Beispiel: T 600 wird als T\_600 oder T600 eingegeben)





**Aufgabe 33:** Deklarieren Sie die komplexe Datenstruktur Drohne in C. Tragen Sie Ihre Lösung direkt in den nachfolgenden Kasten ein. **(max. 10 Punkte)**

```
/* drohne.h */
```

```
#ifndef DROHNE_H_
#define DROHNE_H_
/* laenge eines strings */
#define STR_LEN 20
```

```
void Input_Drohne (Drohne * eineDrohne);
void Output_Drohne (Drohne * eineDrohne);
#endif /* WEGEPUNKT_H_ */
```



**Aufgabe 34:** Die Datenstruktur einer doppelt verketteten Liste **Node** enthält das Element, ein Zeiger auf den nächsten Knoten und ein Zeiger auf den vorherigen Knoten. Erstellen Sie die Datenstruktur Node. Weiterhin wird der Start der Liste durch die globale Variable **begin** vermerkt. Deklarieren Sie diese Variable. Tragen Sie Ihre Lösung direkt in den nachfolgenden Kasten ein. **(max. 7 Punkte)**

```
/* node.h */
```

```
#ifndef NODE_H_
```

```
#define NODE_H_
```

```
Node * Node_New(void);  
void Node_Insert(Node * theNode);  
void Node_Set_Start(Node * theNode);  
void Node_Delete(Node * theNode);  
Node * Node_Get_Start(void);  
#endif /* NODE_H */
```



**Aufgabe 35:** Programmieren Sie die Zugriffsfunktionen Output\_Drohne und Input\_Drohne. Tragen Sie Ihre Lösung direkt in den nachfolgenden Kasten ein. **(max. 16 Punkte)**

**Beschreibung der Funktionen.**

**Output\_Drohne:** Ausgaben der Daten der Struktur Drohne auf die Konsole.

**Input\_Drohne:** Einlesen der Daten der Struktur Drohne von der Konsole.

```
#include <stdio.h>
#include <stdlib.h>
#include "drohne.h"

void Output_Drohne(Drohne * eineDrohne)
{
    printf("Eigenschaften\n");

    printf("Modell: ");
    printf("Hov. Acc. Vertical: ");
    printf("Hov. Acc. Horizontal: ");
    printf("Max Ascent Speed: ");
    printf("Max Descent Speed: ");
    printf("Max Speed: ");
    printf("Max Flight Time: ");
    printf("Max Flight Altitude: ");
}

void Input_Drohne(Drohne * eineDrohne)
{
    printf("Eingabe\n");
    printf("Modell: ");
    scanf(" ");
    printf("Hovering Accuracy Vertical: ");
    scanf(" ");
    printf("Hovering Accuracy Horizontal:");
    scanf(" ");
    printf("Max Ascent Speed: ");
    scanf(" ");
    printf("Max Descent Speed: ");
    scanf(" ");
    printf("Max Speed: ");
    scanf(" ");
    printf("Max Flight Time: ");
    scanf(" ");
    printf("Max Flight Altitude: ");
    scanf(" ");
}
```



**Aufgabe 36:** Vervollständigen Sie die Datei datenbank.c an den markierten Stellen. Programmieren oder ergänzen Sie sinnvoll die Funktionen Eingabe, Ausgabe, Loeschen, Finde und Anzahl. **Verwenden Sie die in drohne.h und node.h deklarierten Funktionen. (max 16 Punkte)**

**Beschreibung der zu ergänzenden oder zu programmierenden Funktionen:**

- **Anzahl** – Gibt die Anzahl der Drohnen in der Datenbank auf die Konsole aus.
- **Finde** – Fragt den Nutzer nach dem Modell einer Drohne, sucht in der Datenbank nach dem Modell und gibt die Daten der Drohne auf die Konsole aus.
- **Eingabe** – Liest Daten der aktuellen Drohne (cursor) von der Konsole ein.
- **Ausgabe** – Gibt die Daten der aktuellen Drohne (cursor) auf der Konsole aus.
- **Löschen** – Löscht die Daten der aktuellen Drohne (cursor).

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "menu.h"
#include "node.h"
Node * cursor;
void Zurueck(void) { }
/***** Menü Bearbeiten *****/
void Eingabe(void) {
}
void Ausgabe(void) {
}
void Loeschen(void) {
}
void Aendern(void) {Ausgabe(); Eingabe();}
struct Menu Bearbeiten_Menu={
    5, "Bearbeiten",
    {
        {"Zurück", Zurueck},
        {"Eingabe", Eingabe},
        {"Ausgabe", Ausgabe},
        {"Löschen", Loeschen},
        {"Ändern", Aendern},
    }
};
/***** Menü Statistik *****/
void Anzahl(void) {
```

```

}

void Finde(void) {

}

struct Menu Statistik_Menu={
    3,"Statistik Menü",
    {
        {"Zurück", Zurueck},
        {"Anzahl Ele.", Anzahl},
        {"Finde...",Finde},
    }
};

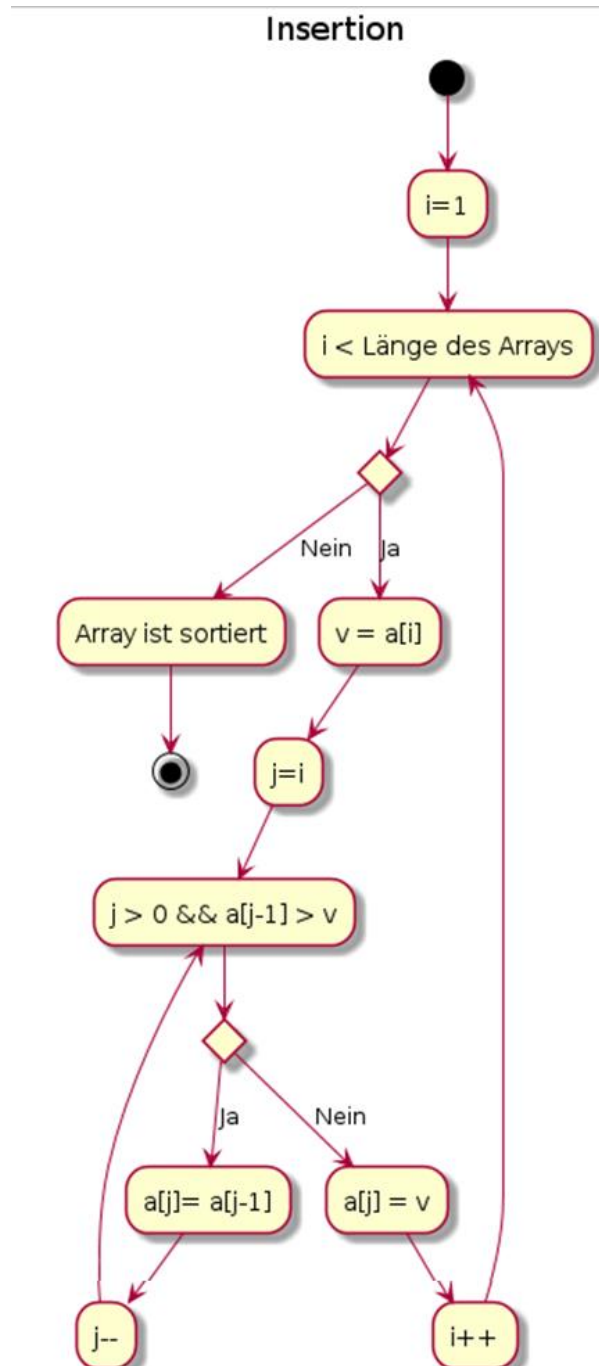
/***** Hauptmenü *****/
void Beenden(void) {exit(0);}
void Bearbeiten_SubMenu(void) {Print_Menu(&Bearbeiten_Menu);}
void Statistik_SubMenu(void) {Print_Menu(&Statistik_Menu);}
struct Menu Main_Menu={
    5,"Hauptmenü",
    {
        {"Beenden", Beenden},
        {"Bearbeiten", Bearbeiten_SubMenu},
        {"Statistik", Statistik_SubMenu},
    }
};

int main(void) {
    cursor = NULL;
    Print_Menu(&Main_Menu);
    return EXIT_SUCCESS;
}

```



**Aufgabe 37:** Gegeben ist folgender Algorithmus in Form eines Activity Diagrams. Der Algorithmus stellt den Sortialgorithmus Insertion Sort aus der Vorlesung dar. Setzen Sie das Diagramm in die Programmiersprache C um. Verwenden Sie dazu die schon angefangene Datei main.c und vervollständigen Sie an der markierten Stelle.  
(max 9 Punkte)



```
/* main.c */

#include <stdio.h>
#include <stdlib.h>

#define ARRAY_SIZE 8
int a[ARRAY_SIZE] = {21,11,1,4,8,14,12,30};
```

```
int main(void) {
    insertion();
    return EXIT_SUCCESS;
}
```



## MATLAB Aufgaben

**Aufgabe 38:** Wie heißt der Befehl, mit dem Sie den Speicher (Workspace) von MATLAB löschen können?

**Aufgabe 39:** Schreiben Sie den MATLAB-Code für die Definition eines Spaltenvektors „w“ mit den Einträgen 8, 4, 2, 6 und 18. Definieren Sie den Vektor so, dass er im Workspace angelegt wird, und im Command Window ausgegeben wird.

**Aufgabe 40:** Bauen Sie eine 5x3 Matrix „Y“ auf, indem Sie drei gegebene 5x1 Zeilenvektoren „a“, „b“ und „c“ transponieren und nebeneinander platzieren. Unterdrücken Sie die Ausgabe im Command Window. Geben Sie den entsprechenden Code an.

**Aufgabe 41:** Es soll eine quadratische 7 x 7 Matrix erzeugt werden, deren obere linke 4x4 Teilmatrix aus einer Diagonalmatrix (mit Einsen auf der Diagonalen) besteht.

- i) Warum schlägt folgender Versuch fehl:  $A = [\text{diag}([1,1,1]), \text{zeros}(4,2); \text{zeros}(2,8)]$
- ii) Korrigieren Sie die Eingabe, sodass die korrekte A-Matrix entsteht.





**Aufgabe 42:** Bauen Sie eine 5x9 Matrix „F“ folgendermaßen auf:

Die Matrix hat 4 Spalten mit jeweils 5 Zeilen, die komplett mit 0'en besetzt sind. Daneben ist ein Block mit 5 Spalten und wieder entsprechend 5 Zeilen, die mit 10'er besetzt sind. Verwenden Sie hierzu die Befehle für die Erzeugung von Nullmatrizen und Matrizen, die mit 1'en besetzt sind.

**Aufgabe 43:** Definieren Sie einen Zeilenvektor „t“, der bei -10 beginnt und in Schritten von 2 bis einschließlich 10 läuft. Unterdrücken Sie die Ausgabe im Command Window.

**Aufgabe 44:** Sie haben einen Vektor „s“ gegeben. Sie wollen von allen Einträgen dieses Vektors elementweise das Quadrat bilden. Benutzen Sie hierzu die Schreibweise „Eintrag hoch 2“. Geben Sie den entsprechenden MATLAB-Code an und nennen Sie den resultierenden Vektor „s2“. Benutzen Sie keine Schleifenprogrammierung, sondern den entsprechenden MATLAB-Befehl für elementweise mathematische Matrix-Operationen.

**Aufgabe 45:** Sie haben eine beliebige Matrix „Y“ gegeben. Sie wollen auf die folgenden Einträge zugreifen: vierte Zeile, Spalten drei bis fünfzehn, und diese Einträge in einen Vektor „r“ abspeichern. Geben Sie den entsprechenden MATLAB-Code an.

**Aufgabe 46:** Sie haben eine beliebige Matrix „X“ gegeben (Einträge sind numerisch, nicht symbolisch). Sie wollen in dieser Matrix den Eintrag (nur diesen!) in der achten Zeile, neunte Spalte durch den Wert „11“ ersetzen. Geben Sie den entsprechenden MATLAB-Code an.



**Aufgabe 47:** Sie haben eine beliebige Matrix „Z“ gegeben (Einträge sind numerisch, nicht symbolisch). Sie wollen in dieser Matrix den Eintrag (nur diesen!) in der achten Spalte, siebten Zeile mit dem 3-fachen seines bisherigen Werts ersetzen. Geben Sie den entsprechenden MATLAB-Code an.

**Aufgabe 48:** Sie wollen die Determinante einer Matrix „C“ berechnen. Geben Sie den entsprechenden MATLAB-Code an.

**Aufgabe 49:** Sie wollen die y-Achse einer bereits vorhandenen Abbildung mit der Beschriftung „Messung [km]“ kennzeichnen. Wie lautet der entsprechende MATLAB-Code?

**Aufgabe 50:** Definieren Sie in einer schon vorhandenen Figure (2d-Plot mit x- und y-Achse) die Achsenabschnitte so, dass auf der x-Achse nur Werte zwischen -20 und -10, auf der y-Achse nur Werte zwischen 5 und 8 angezeigt werden.

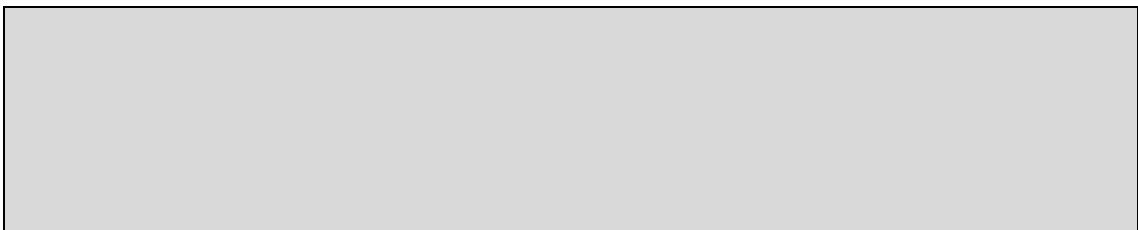
**Aufgabe 51:** Sie wollen den Verlauf einer Kurve im Raum dreidimensional plotten (Komponentenvektoren der Kurve: „p“, „q“, „r“). Wie lautet der zugehörige MATLAB-Code?



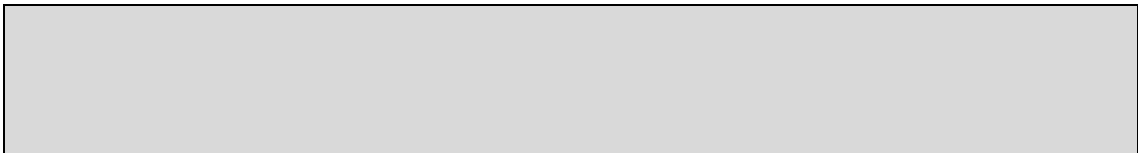
**Aufgabe 52:** Sie wollen den Blickwinkel auf Ihre dreidimensionale Abbildung so einstellen, dass der Blickwinkel eine Elevation von  $85^\circ$  und ein Azimuth von  $-20^\circ$  aufweist. Wie lautet der zugehörige MATLAB-Code?



**Aufgabe 53:** Sie wollen eine Rechenoperation mit Hilfe einer for-Schleife für die Werte  $d$  gleich 10, 55, 10 und -28 umsetzen. Geben Sie den Code für die Schleife an und nennen Sie die in der Schleife auszuführende Rechnung „Rechnung“.



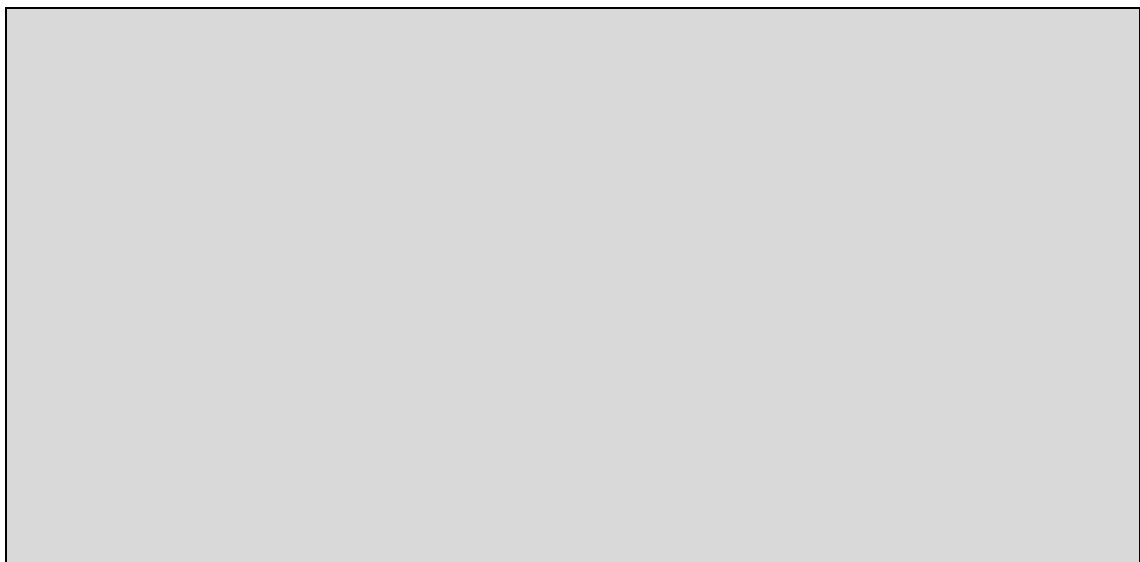
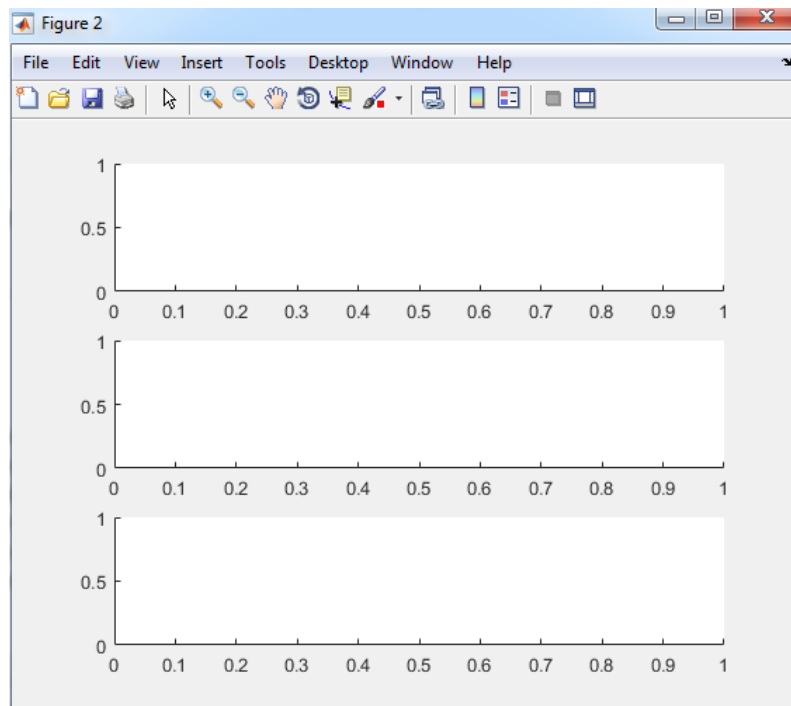
**Aufgabe 54:** Sie wollen allgemein gültige (nicht für einen bestimmten Zahlenwert) Berechnungen für mehrere Variable  $l$ ,  $m$ ,  $n$  und  $o$  durchführen. Geben Sie den MATLAB-Code an, mit dem Sie MATLAB mitteilen, dass es sich bei den Variablen nicht um numerische, sondern um symbolische Variablen handelt.



**Aufgabe 55:** Sie wollen eine gegebene Funktion „ $k$ “ symbolisch nach einer Variable „ $x$ “ integrieren ( $k$  und  $x$  seien bereits als Symbole definiert). Nennen Sie die integrierte Funktion  $K$ . Geben Sie den entsprechenden MATLAB-Code an.



**Aufgabe 56:** Sie wollen drei Vektoren „a“, „b“ und „c“ in eine Abbildung (eine Figure), aber in drei untereinander liegende Teilabbildungen plotten (siehe Abbildung). Geben Sie den entsprechenden Code an. Erzeugen Sie zunächst eine neue Abbildung (Figure). Erzeugen Sie dann eine obere Teilabbildung und bilden Sie den Vektor „a“ in dieser oberen Teilabbildung ab. Erzeugen Sie nun die mittlere Teilabbildung, und stellen Sie den Vektor „b“ in der mittleren Teilabbildung dar. Bilden Sie den Vektor „c“ in der unteren Teilabbildung ab.

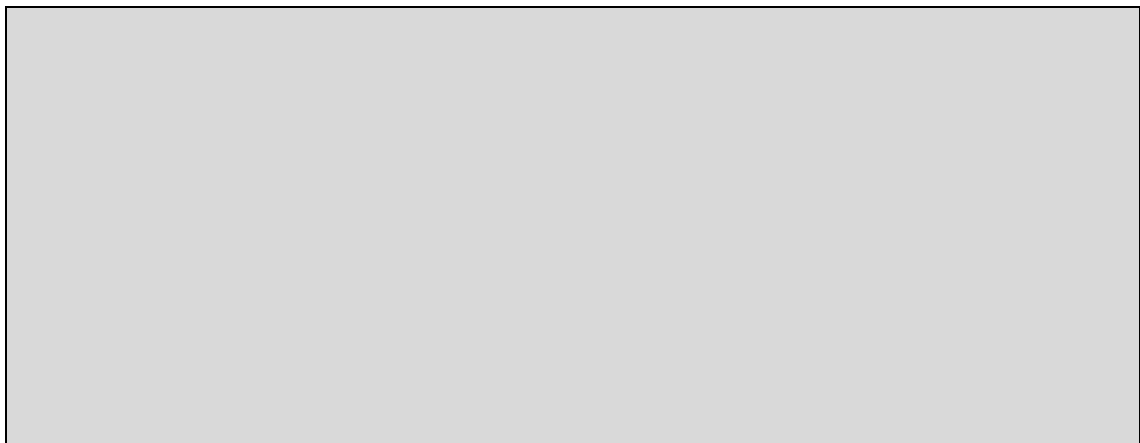


**Aufgabe 57:** Sie haben den gemessenen Positionsverlauf eines Flugzeugs in einer Messmatrix „Messung“ gegeben. In dieser Messmatrix sind die Messungen für die Nord-, Ost- und Höhe-Positionswerte als Zeilenvektoren abgespeichert:

1. Zeile: Positionsverlauf in Nord-Richtung
2. Zeile: Positionsverlauf in Ost-Richtung
3. Zeile: Positionsverlauf in Höhen-Richtung

Plotten Sie den Positionsverlauf in einer 3d-Darstellung. Gehen Sie wie folgt vor:

- Öffnen Sie zunächst eine neue, leere Abbildung.
- Erzeugen Sie die Messvektoren „Nord“, „Ost“ und „Hoehe“ aus der Matrix „Messung“.
- Plotten Sie den Bewegungsverlauf in drei Dimensionen. Linienfarbe „schwarz“, Lini-  
enart „gestrichelt“ („dashed“, --).
- Fügen Sie aus Gründen der Anschaulichkeit ein Gitter in die Abbildung ein.



**Aufgabe 58:** Sie sollen mit Hilfe des Befehls „fill3“ eine Fläche in MATLAB visualisieren. Die Fläche ist durch die folgenden Eckpunkte (jeweils x,y und z-Koordinaten) definiert:

Eckpunkt 1: [1 1 1]

Eckpunkt 2: [0 1 2]

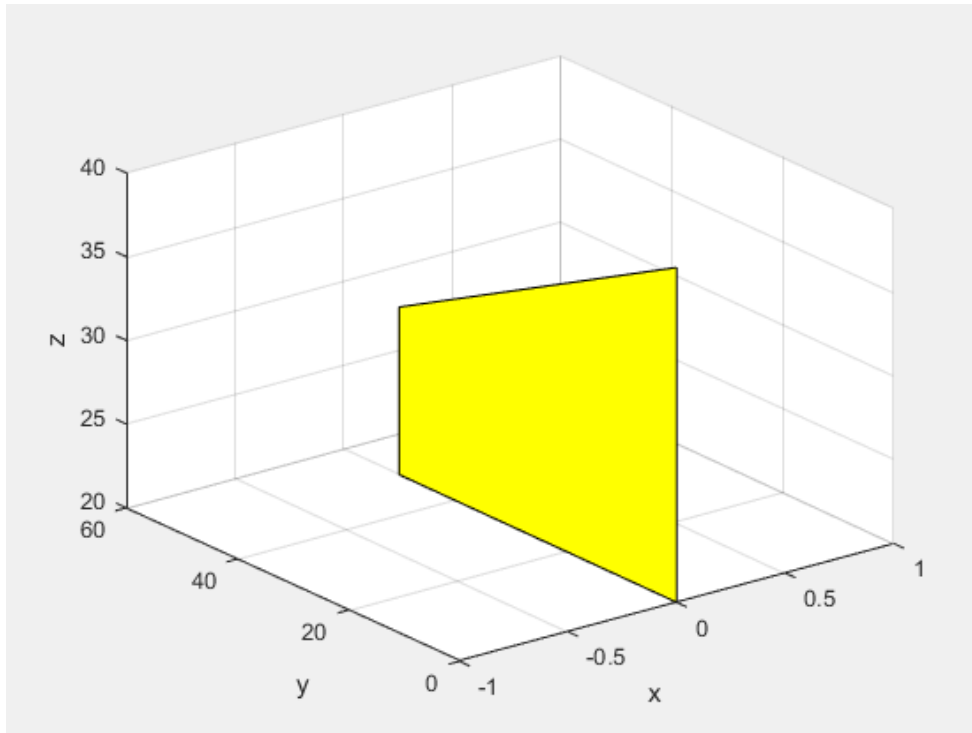
Eckpunkt 3: [0 1 3]

Stellen Sie mit Hilfe des Befehls „fill3“ diese Fläche dar. Farbe: blau (Hilfe: ‚b‘).





**Aufgabe 59:** Welcher der folgenden Codes erzeugt die in der Abbildung gezeigte Fläche?



| Code a:                                                                                                                              | Code b:                                                                                                                              | Code c:                                                                                                                              |
|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------|
| <pre>X = [0 0; 0 0] Y = [0 0; 50 50] Z = [20 20; 40 30] surf(X,Y,Z) hold colormap([1 1 0]) xlabel('x') ylabel('y') zlabel('z')</pre> | <pre>X = [0 0; 0 0] Y = [0 0; 50 50] Z = [20 40; 20 30] surf(X,Y,Z) hold colormap([1 1 0]) xlabel('x') ylabel('y') zlabel('z')</pre> | <pre>X = [0 0; 0 0] Y = [0 0; 50 50] Z = [20 30; 20 40] surf(X,Y,Z) hold colormap([1 1 0]) xlabel('x') ylabel('y') zlabel('z')</pre> |

- ☐ Code a:  
☐ Code b:  
☐ Code c:



**Aufgabe 60:** Sie haben die folgende for-Schleife gegeben:

```
for k = 1:1:5
    for l = 1:1:3
        Ergebnis = k*11 + l*3;
        Ergebnis_Mat(k,l) = Ergebnis;
    end
end
Ergebnis_Mat
```

Welche finale Ausgabe erwarten Sie bei der Ausführung dieses Codes?

| a)                                                                                                       | b)                                                                                     | c)                                                                                                       |
|----------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------|
| Ergebnis_Mat =<br>14    17    20<br>25    28    31<br>36    39    42<br>47    50    53<br>58    61    64 | Ergebnis_Mat =<br>14    17    20<br>25    28    31<br>36    39    42<br>47    50    53 | Ergebnis_Mat =<br>13    15    17<br>24    26    28<br>35    37    39<br>46    48    50<br>57    59    61 |

- ☐ Ausgabe a)  
☐ Ausgabe b)  
☐ Ausgabe c)

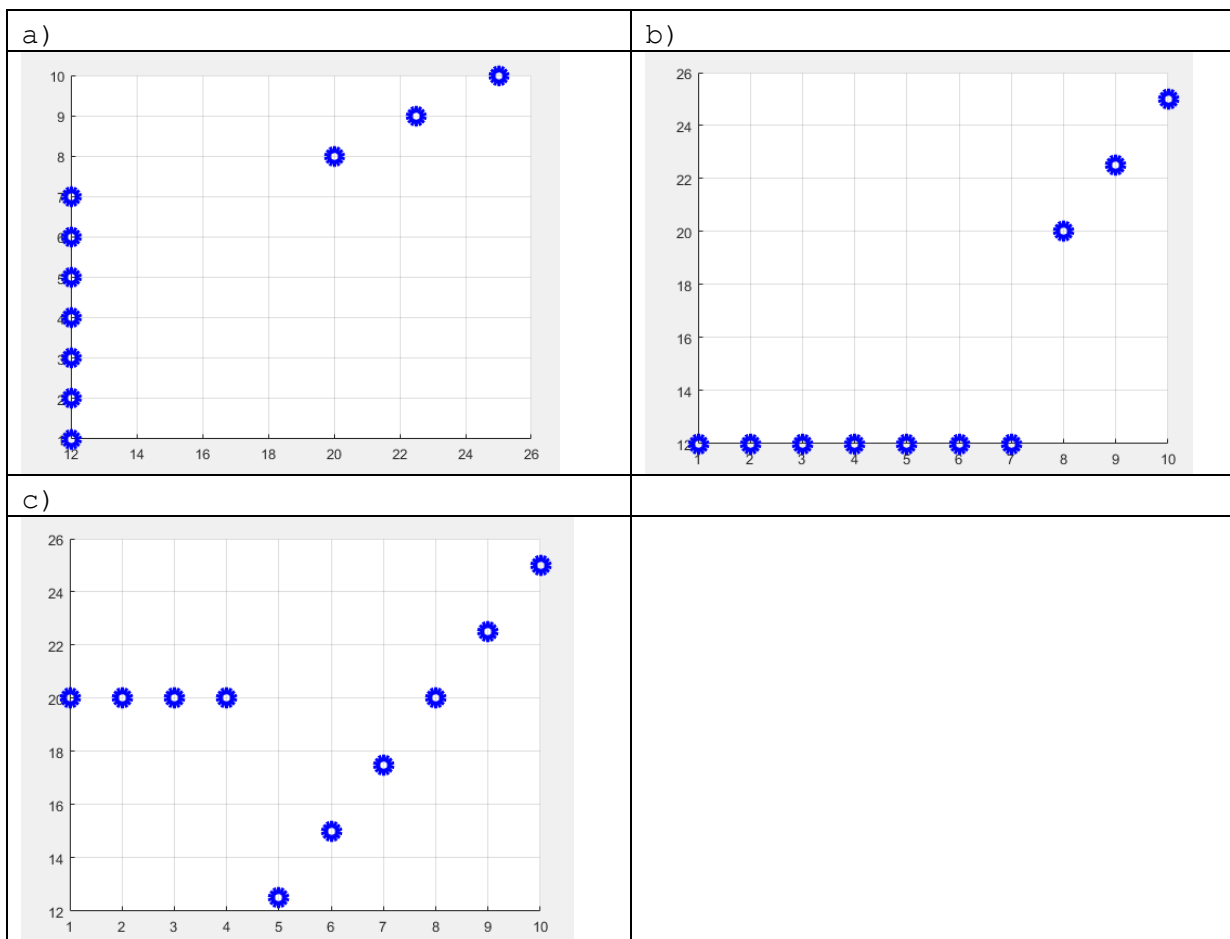


**Aufgabe 61:** Sie haben den folgenden MATLAB-Code gegeben:

```
figure
hold
grid
k = 0

while k < 10
    k = k + 1
    Ergebnis = k * 2.5
    if Ergebnis < 20
        Ergebnis = 12
    else
        end
    plot(k,Ergebnis, 'bo', 'Markersize', 10, 'Linewidth', 5)
end
```

Welche der drei folgenden Abbildungen wird durch diesen Code erzeugt?



- ☐ Abbildung a)
- ☐ Abbildung b)
- ☐ Abbildung c)



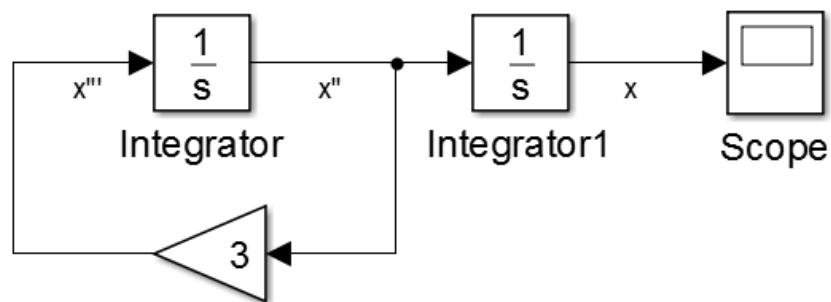


**Aufgabe 62:** Sie haben die Differentialgleichung eines Wägelchens mit Dämpfung  $d$  gegeben:

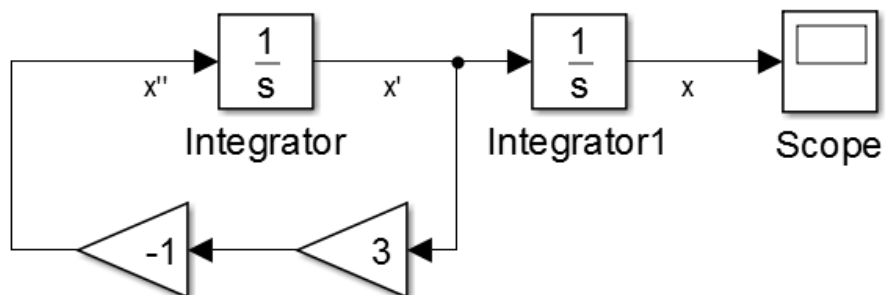
$$x''' + 3 * x'' = 0$$

Welches der dargestellten Simulink - Blockschaltbilder bildet diese Differentialgleichung ab?

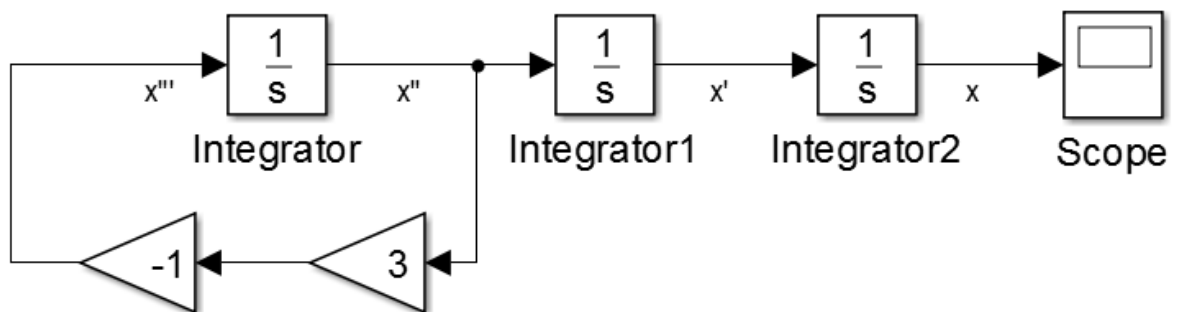
a)



b)



c)

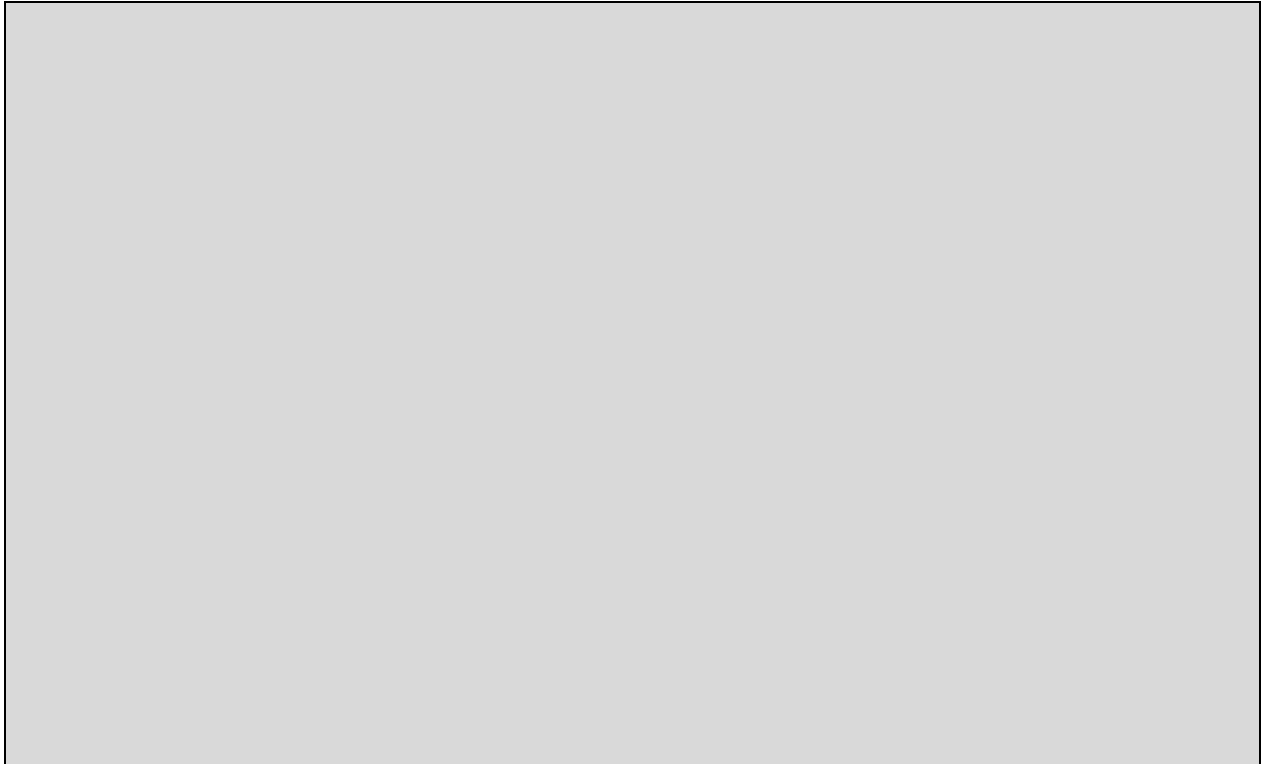


- ☐ Blockschaltdiagramm a)
- ☐ Blockschaltdiagramm b)
- ☐ Blockschaltdiagramm c)



**Aufgabe 63:** Zeichnen Sie ein Simulink-Blockschaltbild, das die folgende Differentialgleichung abbildet:

$$x''' - 8 * x'' + 2 * x' + 4 * x = 0$$



## C-Funktionen

**FILE \*fopen(const char \*filename, const char \*mode)**

fopen eröffnet die angegebene Datei und liefert einen Datenstrom oder NULL bei Mißerfolg.

**int fclose(FILE \*stream)**

fclose schreibt noch nicht geschriebene Daten für stream, wirft noch nicht gelesene, gepufferte Eingaben weg, gibt automatisch angelegte Puffer frei und schließt den Datenstrom

**int printf(const char \*format, ...)**

printf wandelt Ausgaben um und schreibt sie in **stdout** unter Kontrolle von **format**.

**int scanf(const char \*format, ...)**

scanf liest von **stdin** unter Kontrolle von **format** und legt umgewandelte Werte mit Hilfe von nachfolgenden Argumenten (...) ab

**int fgetc(FILE \*stream)**

fgetc liefert das nächste Zeichen aus stream als unsigned char (umgewandelt in int)

**int getchar(void)**

getchar ist äquivalent zu `getc(stdin)`.

**char \*gets(char \*s)**

gets liest die nächste Zeile von stdin in den Vektor s und ersetzt dabei den abschließenden Zeilentrenner durch `'\0'`.

**int putchar(int c)**

putchar(c) schreibt das Zeichen c (umgewandelt in unsigned char) in stdout.

**int atoi(const char \*s)**

atoi wandelt s in int um

**void \*malloc(size\_t size)**

malloc liefert einen Zeiger auf einen Speicherbereich für ein Objekt der Größe size oder NULL

**void free(void \*p)**

free gibt den Bereich frei, auf den p zeigt

**void exit(int status)**

exit beendet das Programm normal

**int strcmp(char \*cs, char \*ct)**

Zeichenketten **cs** und **ct** vergleichen; liefert <0 wenn **cs** < **ct**, 0 wenn **cs** == **ct**, oder >0, wenn **cs** > **ct**.

**size\_t strlen(char \*cs)**

liefert die Länge von **cs** (ohne `'\0'`).

**char \*strcat(char \*s, char \*ct)**

Zeichenkette **ct** wird hinten an die Zeichenkette **s** angefügt und gibt s zurück

