

**Sommersemester 2021**

## **Ingenieurinformatik, Teil 1: C-Programmierung**

**Schriftliche Fernprüfung mit Videoaufsicht**

**Prüfer: Selting, Ressel, Küpper, Reichl und KollegInnen**

**Bearbeitungszeit: 60 Minuten**

**Hilfsmittel:**

- Taschenrechner und elektronische Hilfsmittel sind nicht zugelassen.
- Alle schriftlichen Unterlagen sind erlaubt.
- Der PC darf während der Prüfung nur zur Anzeige des Aufgabenblatts genutzt werden.

**Schreiben Sie Ihren Namen, Vornamen und auch die Studiengruppe auf alle Lösungsblätter. Es werden nur handschriftliche Lösungen auf leeren, weißen DIN-A4-Blättern akzeptiert.**

**Wenn Sie zur Kombiprüfung „Ingenieurinformatik“ angemeldet sind, dann beachten Sie bitte, dass Sie beide Teile (C-Programmierung und Numerik/ Matlab) im selben Semester schreiben müssen.**

**\*\*\* Viel Erfolg! \*\*\***

### Aufgabe 1 (allg. Programmierung, ca. 24 Punkte)

Eine Masse wird unter dem Winkel  $\varphi$  zur x-Achse mit einer Anfangsgeschwindigkeit

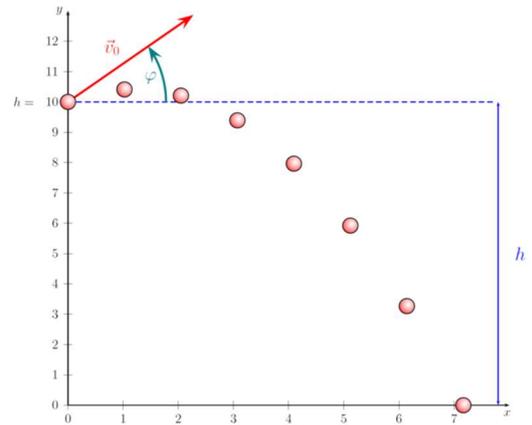
$$\vec{v}_0 = \begin{pmatrix} v_{x,0} \\ v_{y,0} \end{pmatrix} = \begin{pmatrix} 5 \cos(\varphi) \\ 5 \sin(\varphi) \end{pmatrix} \left[ \frac{m}{s} \right]$$

aus einer Höhe  $h = 10$  Meter abgeworfen.

Beim sog. schiefen Wurf ohne Reibung gelten für die zeitabhängigen Koordinaten  $x(t)$ ,  $y(t)$  des Massenschwerpunktes die Gleichungen

$$x(t) = v_{x,0} \cdot t$$

$$y(t) = h + v_{y,0} \cdot t - \frac{1}{2} \cdot g \cdot t^2 \quad \text{mit der Erdbeschleunigung } g = 9,81 \left[ \frac{m}{s^2} \right]$$



Die Abbildung oben zeigt die Masse (rote Kugel) während des Wurfs zu verschiedenen Zeitpunkten im x-y-Koordinatensystem. Die Masse kommt zum Zeitpunkt  $t_{End}$  auf dem Boden auf d.h.  $y(t)=0$ :

$$t_{End} = \frac{v_{y,0}}{g} + \sqrt{\left(\frac{v_{y,0}}{g}\right)^2 + \frac{2h}{g}} \quad [s]$$

Schreiben Sie ein C-Programm, welches die folgenden Aufgaben löst:

- Beim Start wird der Anwender nach dem Abwurf-Winkel  $\varphi$  im **Gradmaß** gefragt, mit der Vorgabe  $10^\circ \leq \varphi \leq 35^\circ$ . Falls die Winkel-Eingabe nicht im Bereich zwischen 10 und 35 Grad liegt, soll eine Fehlermeldung ausgegeben und die Eingabe wiederholt werden, bis ein Winkel aus diesem Bereich eingegeben wurde.
- Das Zeitintervall  $[0, t_{End}]$  soll in 30 Teilintervalle  $[t_i, t_{i+1}]$  aufgeteilt werden  $t_0 = 0, t_1, t_2, \dots, t_{30} = t_{End}$  und die Zeitpunkte  $t_i$  (in Sekunden) sowie  $x(t_i)$  und  $y(t_i)$  (in Metern) sollen jeweils mit **vier Nachkommastellen** tabellarisch ausgegeben werden (siehe nachfolgende Abbildung).
- Kurz vor dem Ende des Programms, nachdem (!) die Tabelle vollständig auf dem Bildschirm ausgegeben wurde, werden die folgenden Informationen ausgegeben:
  1. Die maximale Höhe  $h_{max}$ , die bei  $t_{max} \in \{t_0, t_1, \dots, t_{30}\}$  erreicht wurde und auch der Zeitpunkt  $t_{max}$  (jeweils **zwei Nachkommastellen**).
  2. Die zur maximalen Höhe  $h_{max}$  gehörigen x- und y-Koordinaten  $x_{max} = x(t_{max})$  und  $y_{max} = y(t_{max})$  (jeweils **zwei Nachkommastellen**).

Die Bildschirmausgabe Ihres Programms soll dem folgenden Bildschirmfoto entsprechen:

```

C:\Q2\Q25.14.2\Tools\QICreator\bin\qcreator_process_stub.exe
Geben Sie den Winkel phi in Grad ein ( 10<=phi<=35 ): 40
Falsche Eingabe fuer phi: 10<=phi<=35
Geben Sie den Winkel phi in Grad ein ( 10<=phi<=35 ): 30
Benutzereingabe

t          x(t)          y(t)
-----
0.0000    0.0000        10.0000
0.0568    0.2461        10.1263
0.1137    0.4923        10.2208
0.1705    0.7384        10.2837

      ⋮              ⋮              ⋮

1.5347    6.6456        2.2837
1.5916    6.8917        1.5541
1.6484    7.1378        0.7929
1.7052    7.3839        0.0000

Die maximale Hoehe 10.28 Meter wird nach 0.17 Sekunden erreicht.
Die zugehoerigen Koordinaten sind xmax= 7.38 und ymax=10.28
    
```

## Aufgabe 2 (Struktogramm, ca. 9 Punkte)

In einem C-Programm sind die Matrix (Preistabelle: **pt**) sowie der Preis-Vektor (Einzelpreis: **ep**) als globale Variablen definiert:

```
double pt[10][4], ep[10];
```

Ein Schraubenhersteller verkauft 10 verschiedene Typen von Schrauben (Typ 1-10).

Der Einzelpreis für eine Packung Schrauben vom Typ *k* steht als *k*-tes Element im Vektor **ep**.

Für Bestellungen bis 5 Packungen eines Typs gilt der jeweilige Einzelpreis. Bei einer Bestellung von 6 oder mehr Packungen wird ein Rabatt pro Packung wie in untenstehender Tabelle gewährt:

Packungsanzahl	1-5	6-10	11-20	>20
Rabatt für alle Preiskategorien	0%	3%	7%	15%

Es soll eine Funktion mit dem Namen „belegen“ programmiert werden. Diese Funktion schreibt die folgenden Werte in die Matrix „**pt**“:

- In der 1. Spalte steht für Bestellungen bis 5 Packungen der Einzelpreis (=Vektor „**ep**“),
- in der 2. Spalte steht für Bestellungen von 6 bis 10 Packungen der rabattierte Preis,
- in der 3. Spalte steht für Bestellungen von 11 bis 20 Packungen der rabattierte Preis,
- in der letzten Spalte steht für Bestellungen über 20 Packungen der rabattierte Preis

In jeder Spalte steht der (rabattierte) Preis für jeden Typ 1-10. <sup>1</sup>

Typ	0-5 Stück	6-10 Stück	11-20 Stück	ab 21 Stück
1	1.25	1.21	1.16	1.06
2	1.70	1.65	1.58	1.44
⋮	⋮	⋮	⋮	⋮
10	3.25	3.15	3.02	2.76

Vektor **ep** (links) zeigt auf die erste Spalte. Matrix **pt** (rechts) zeigt auf die gesamte Matrix.

Zeichnen Sie ein Struktogramm der Funktion „belegen“. Alle Kontrollstrukturen (**2 verschachtelte Schleifen, 1 Mehrfachverzweigung**) müssen klar erkennbar und korrekt gezeichnet sein.

**Hinweis: In dieser Aufgabe 2 soll kein C-Quelltext geschrieben werden!**

<sup>1</sup> Die Preise in der Ausgabe der Beispielmatrix **pt** sind auf 2 Nachkommastellen gerundet.

### Aufgabe 3 (Funktionen, Felder, ca. 10 Punkte)

In einem C-Programm sind alle Variablen als lokale Variablen definiert. Es soll eine Funktion „skalarprod“ mit der Deklaration

```
double skalarprod(double *x, double *y, int n);
```

programmiert werden, welche das Skalarprodukt  $\langle \vec{x}, \vec{y} \rangle$  der Vektoren  $\vec{x}, \vec{y} \in \mathbb{R}^n$  berechnet.

In einer Formelsammlung finden Sie z. B.:

$$\langle \vec{x}, \vec{y} \rangle := \sum_i x_i y_i$$

- a) Schreiben Sie den C-Quelltext der Funktion „skalarprod“. Diese Funktion soll das Skalarprodukt zweier beliebiger Vektoren  $\vec{x}, \vec{y} \in \mathbb{R}^n$  in einer Schleife berechnen und das Ergebnis als Rückgabewert an das aufrufende Programm liefern.

**Hinweis: In dieser Teilaufgabe 3a) soll kein Struktogramm gezeichnet werden!**

- b) Wie lautet der Funktionsaufruf, in der unten fehlenden Zeile, wenn hier mit Hilfe der Funktion „skalarprod“ aus Teilaufgabe 3a) der Wert der Variablen `norm2` gemäß der Formel

$$\text{norm2} = \langle \vec{x}, \vec{x} \rangle$$

berechnet werden soll.

```
1  #include <stdio.h>
2  #include <math.h>
3  double skalarprod(double *x, double *y, int n);
4
5  int main(void)
6  {
7      double x[] = {1.3, 2.0, 3.7, 4.1, 5.9, 6.1, 7.6, 8.8, 9.3, 10.0};
8      double norm2;
9
10     

11         Hier kommt der Funktionsaufruf von „skalarprod“
12


13
14     ⋮
15 }
```

#### Aufgabe 4 (Fehlersuche, ca. 10 Punkte)

Das folgende Programm belegt zwei Vektoren mit Zufallszahlen aus dem Bereich [5.0, 8.0] und berechnet anschließend das aus der Mathematik bekannte Vektor- oder Kreuzprodukt.

In einer Formelsammlung finden Sie z. B.:

$$\vec{u} \times \vec{v} = \begin{pmatrix} u_1 \\ u_2 \\ u_3 \end{pmatrix} \times \begin{pmatrix} v_1 \\ v_2 \\ v_3 \end{pmatrix} = \begin{pmatrix} u_2 v_3 - u_3 v_2 \\ u_3 v_1 - u_1 v_3 \\ u_1 v_2 - u_2 v_1 \end{pmatrix}$$

Im Quelltext befinden sich **fünf Fehler**.

- In welchen Zeilen befinden sich die Fehler?
- Geben Sie jeweils die korrekte Version der kompletten (!) Zeile an.

```
1 #include <stdio.h>
2
3 #define DIM 3
4
5 int main()
6 {
7     float x[DIM], y[DIM], erg[DIM];
8     int i = 0;
9     while(i <= DIM)
10    {
11        x[i] = 3.0*rand()/RAND_MAX+8.0;
12        y[i] = 3.0*rand()/RAND_MAX+5.0;
13    }
14
15    //
16    printf("Die Zufallszahlen liegen ");
17    printf("im Intervall [%d,%d]\n", 3, 8);
18    //
19    erg[0] = x[1]*y[2]-x[2]*y[1];
20    erg[1] = x[2]*y[0]-x[0]*y[3];
21    erg[2] = x[0]*y[1]-x[1]*y[0];
22    //
23    printf("%9.4f      %9.4f      %9.4f\n", x[0], y[0], erg[0]);
24    printf("%9.4f x  %9.4f =  %9.4f\n", x[1], y[1], erg[1]);
25    printf("%9.4f      %9.4f      %9.4f\n\n", x[2], y[2], erg[2]);
26    return 0;
27 }
```

### Aufgabe 5 ( Operatoren, Bytes, Zeichenketten, ca. 14 Punkte)

4.1. Beantworten Sie zu folgendem Quelltext-Ausschnitt die Fragen a) – d)

```
5 char text[11] = "SoSe2021";
6 printf("%s \n", text);
7 text[6] = '\0';
8 printf("%s \n", text);
9 printf("%d Zeichen \n", (int) strlen(text));
```

- Wieviel Byte Speicherplatz benötigt die Variable text ?
- Welche Programmausgabe erzeugt das obige Quelltext-Teilstück in Zeile 6 ?
- Welche Programmausgabe erzeugt das obige Quelltext-Teilstück in Zeile 8 ?
- Welche Programmausgabe erzeugt das obige Quelltext-Teilstück in Zeile 9 ?

4.2. Welche Werte haben die Variablen v und w nach Ausführen der nachfolgenden Quelltextzeilen:

```
11 int x,y,v,w;
12 x = 2; y = 3;
13 v = x++ + y++;
14 w = ++x + ++y;
```

Begründen Sie die Berechnung Ihrer Werte.

4.3. Zur 2-Byte short Variablen i wird der dezimale Wert 3 addiert:

```
short i = 32767;
i = i + 3;
```

Welchen Wert hat die Variable i nach der Addition? Begründen Sie Ihre Antwort.  
(Hinweis:  $32767 = 2^{15}-1$ )

4.4. Unterscheidet sich der Speicherplatzbedarf der Variablen x und y wenn diese wie folgt definiert sind:

```
float *x;
double *y;
```

Begründen Sie Ihre Antwort.