

Ingenieurinformatik

C-Programmierung

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Bachelorstudiengang:

- Studienbeginn vor WS13/14 (Kombinationsprüfung) **
- Studienbeginn ab WS13/14 bis WS15/16 **
- Studienbeginn ab SS16 (Kombinationsprüfung)

- Diplomstudiengang Fahrzeugtechnik **

**** Die Prüfung ist nur dann gültig, wenn Sie die Zulassungsvoraussetzung erworben haben (erfolgreiche Teilnahme am Praktikum).**

Aufgabensteller: Dr. Reichl, Dr. Küpper und Kollegen

Bearbeitungszeit: 60 Minuten

Hilfsmittel: Taschenrechner nicht zugelassen,
PC/Notebook nicht zugelassen,
sonstige eigene Hilfsmittel sind erlaubt,
Bearbeitung mit Bleistift ist erlaubt.

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

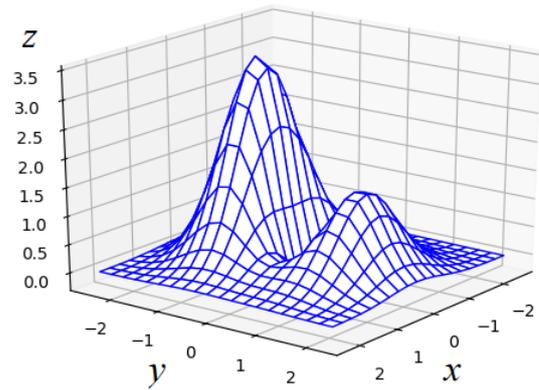
Aufgabe 1: (ca. 23 Punkte)

Die Funktion $f_b(x, y)$ hängt von den beiden Veränderlichen x und y sowie von dem Parameter b ab:

$$f_b(x, y) = (b \cdot x^2 + \frac{5}{2} \cdot y^2 - y) \cdot e^{1-(x^2+y^2)}$$

Schreiben Sie ein C-Programm, welches die folgenden Aufgaben löst (siehe Bildschirmfoto unten!):

- Direkt nach dem Programmstart muss der Anwender einen Wert für den Parameter b eingeben.
- Der Parameter b muss im Bereich $-5 \leq b \leq +5$ liegen, **andernfalls wird eine Fehlermeldung angezeigt** und die Eingabe wird wiederholt.
- Das Programm ermittelt den maximalen Funktionswert $f_b(x, y)$ im Bereich $-2,5 \leq x \leq +2,5$ und $-2,5 \leq y \leq +2,5$. Das Programm soll dazu alle x - und y -Werte von $-2,5$ bis $+2,5$ jeweils mit einer Schrittweite von $0,05$ durchlaufen, überall die Funktion $f_b(x, y)$ ausrechnen und dabei den maximalen Funktionswert suchen.
- Anschließend werden der maximale Funktionswert sowie die dazugehörigen x - und y -Koordinaten wie im Bildschirmfoto gezeigt mit drei Nachkommastellen ausgegeben.
- Wenn der maximale Funktionswert größer als 4 ist, wird die Meldung „Maximum > 4.0“ ausgegeben, ansonsten wird die Meldung „Maximum <= 4.0“ ausgegeben.
- **Die Berechnung des Funktionswerts $f_b(x, y)$ soll nicht im Hauptprogramm `main()` sondern in einer separaten Unterfunktion erfolgen.**



Der Parameter b muss im Bereich $-5...+5$ liegen, fehlerhafte Eingaben müssen wiederholt werden.

```
C:\Qt\Tools\QtCreator\bin\qtcreator_process_stub.exe
b: -10
Eingabefehler!
b: 123
Eingabefehler!
b: 0.5
Maximum bei x = 0.000; y = -0.900; f(x,y) = 3.537
Maximum <= 4.0
```

Nach jeder ungültigen Eingabe wird eine Fehlermeldung angezeigt!

Hier wurde ein gültiger Wert für b eingegeben.

Aufgabe 2: (ca. 19 Punkte)

Das folgende C-Programm dient zur Steuerung eines Kühlschranks. Die Funktion `get_temp()` liefert die aktuelle Temperatur (in °C) im Kühlschrank als `double`-Wert. Die Funktion `set_comp()` dient zum Ein- bzw. Ausschalten des Kühlschrank-Kompressors. **Die genauen Definitionen dieser beiden Funktionen sind nicht abgedruckt, sie können als gegeben vorausgesetzt werden!**

```
#include <stdio.h>
#define TEMP1 6.0
#define TEMP2 2.0

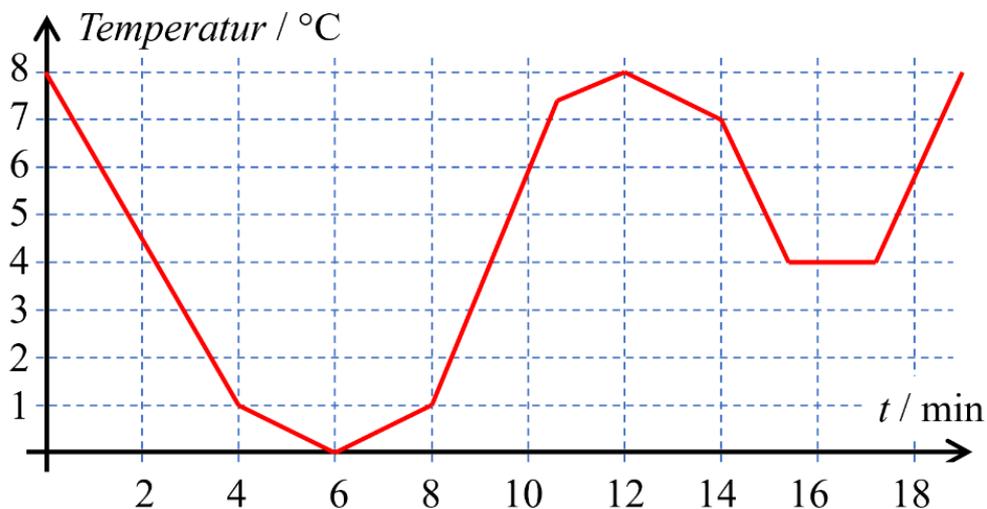
/* Aktuelle Temperatur im Kühlschrank ermitteln */
double get_temp(void);

/* on_off == 1: Kühlschrank-Kompressor einschalten */
/* on_off == 0: Kühlschrank-Kompressor ausschalten */
void set_comp(int on_off);

int main(void)
{
    float temp;
    int state = 0;

    while(1 == 1)
    {
        temp = get_temp();
        if(temp > TEMP1)
        {
            if(state == 0) { set_comp(1); state = 1; }
        }
        if(temp < TEMP2)
        {
            if(state == 1) { set_comp(0); state = 0; }
        }
    }
    return 0;
}
```

2.1. Das Diagramm zeigt den zeitlichen Verlauf der Temperatur im Kühlschrank. **Markieren Sie im Diagramm die Zeitintervalle**, in denen der Kühlschrank-Kompressor eingeschaltet ist.



- 2.2. Das C-Programm soll schon einmal getestet werden, obwohl der Kühlschrank noch gar nicht aufgebaut worden ist. Zu diesem Zweck wird die Funktion `get_temp()` durch die folgende Variante ersetzt: Dadurch werden simulierte (zufällige) Temperaturwerte erzeugt.

```
#include <stdlib.h>
double get_temp(void) /* Wird nur zum Testen eingesetzt! */
{
    double simul = 1 + (double)rand() / RAND_MAX * 6;
    return simul;
}
```

Welches ist der minimale bzw. maximale (simulierte) Temperaturwert, der von der abgebildeten Funktion `get_temp()` als Rückgabewert zurückgegeben werden kann?

Minimaler Temperaturwert: °C Maximaler Temperaturwert: °C

- 2.3. Zeichnen Sie ein Struktogramm, welches den genauen Ablauf der Funktion `main()` beschreibt. Alle im C-Quelltext vorhandenen Schleifen und Verzweigungen sollen erkennbar sein.

Aufgabe 3: (ca. 18 Punkte)

- 3.1. Im abgebildeten C-Quelltext ist die Funktion `char2num()` definiert (Seite 7, oben). Welchen Rückgabewert liefert diese Funktion für die folgenden Aufrufe?

`char2num('5');` → Rückgabewert:

`char2num('9');` → Rückgabewert:

- 3.2. Fügen Sie die Definition der Funktion `scan_date()` zum abgebildeten C-Quelltext hinzu (Seite 7, Mitte). An die Funktion `scan_date()` wird eine Zeichenkette im Format „dd.mm.“ übergeben (es handelt sich dabei um eine Datumsangabe mit `dd` = Nummer des Tages, `mm` = Nummer des Monats). Aus der Zeichenkette sollen die Nummern des Tages und des Monats ermittelt und mithilfe von Zeiger-Parametern zurückgegeben werden.

Tipps: - Im Hauptprogramm sind zwei Beispiele abgedruckt, wie die Funktion `scan_date()` aufgerufen werden kann und welche Ergebnisse sie jeweils zurückgibt.

- Sie dürfen voraussetzen, dass die übergebene Zeichenkette korrekt formatiert ist.

- 3.3. Beschreiben Sie eine Möglichkeit, wie die Funktion `scan_date()` auch ohne Verwendung von Zeiger-Parametern implementiert werden könnte. (Keinen C-Quelltext angeben, sondern mit einigen Stichworten beantworten!)

Wie würde in diesem Fall die Funktionsdeklaration von `scan_date()` aussehen?

```
#include <stdio.h>
int char_to_num(char ch);
void scan_date(char *str, int *day, int *month);

/* Hauptprogramm */
int main(void)
{
    int d, m;

    /* Tag und Monat in einer Zeichenkette finden */
    scan_date("30.01.", &d, &m);
    printf("d = %d, m = %d\n", d, m); /* d = 30, m = 1 */

    scan_date("24.12.", &d, &m);
    printf("d = %d, m = %d\n", d, m); /* d = 24, m = 12 */
    return 0;
}
```

```

/* Aufgabe 3.1 */
int char_to_num(char ch)
{
    return ch - '0';
}

/* Aufgabe 3.2 */
void scan_date(char *str, int *day, int *month)
{

```

```

}

```

- 3.4. Das folgende C-Programm dient dazu, die Großbuchstaben in einer Zeichenkette zu zählen. Im C-Quelltext befinden sich fünf Fehler, korrigieren Sie diese Fehler!

```

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

int count_upper(char *str);

int main(void)
{
    char my_str[];
    printf("Text eingeben: ");
    scanf("%99s", my_str);
    printf("%d Großbuchstaben gefunden.\n", count_upper(my_str));
    return 0;
}

int count_upper(char *str);
{
    int i, count = 0;
    for(i = 0; i < (int)strlen(str); i = i--)
    {
        if(str[i] >= 'A', str[i] <= 'Z')
            ++count;
    }
    return count;
}

```

Aufgabe 4: (ca. 7 Punkte)

Geben Sie zu der Dezimalzahl **-43** die 8-Bit-Zweierkomplementdarstellung an. Notieren Sie alle Schritte Ihrer Berechnung auf dem Aufgabenblatt.