

Ingenieurinformatik

C-Programmierung

Name	Vorname	Semestergruppe	Hörsaal

	Aufgabe 1	Aufgabe 2	Aufgabe 3	Aufgabe 4	Aufgabe 5	Summe

Beginn Ihres Bachelorstudiums:

- vor WS13/14 (Kombiprüfung C + MATLAB) **
- von WS13/14 bis WS15/16 **
- von SS16 bis WS17/18 (Kombiprüfung C + MATLAB)
- ab SS18

**** 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. 25 Punkte)

Schreiben Sie ein C-Programm, welches das bestimmte Integral der Funktion $f(x) = (\sin x)^2$ auf dem Intervall $[a; b]$ mit dem „Monte-Carlo-Verfahren“ numerisch ermittelt:

1. Zwei Zählervariablen $z1$ und $z2$ werden definiert und zunächst auf 0 gesetzt.
2. Der Anwender gibt die Integrationsgrenzen a und b ein.
3. Die folgenden Schritte 3.1. bis 3.3. werden in einer Schleife 1000-fach wiederholt. Auf den Aufruf der Funktion `rand()` zur Initialisierung des Zufallsgenerators kann verzichtet werden:
 - 3.1. Es wird ein zufälliger x -Wert (Datentyp „double“) im Bereich $a \dots b$ erzeugt.
 - 3.2. Es wird ein zufälliger y -Wert (Datentyp „double“) im Bereich $0 \dots 1$ erzeugt.
 - 3.3. Falls $f(x) < y$ ist, dann wird der Zähler $z1$ inkrementiert, ansonsten wird $z2$ inkrementiert.
4. Anschließend wird der Näherungswert f_n für das gesuchte Integral wie folgt ermittelt:

$$f_n = \frac{b - a}{z1 + z2} \cdot z2$$

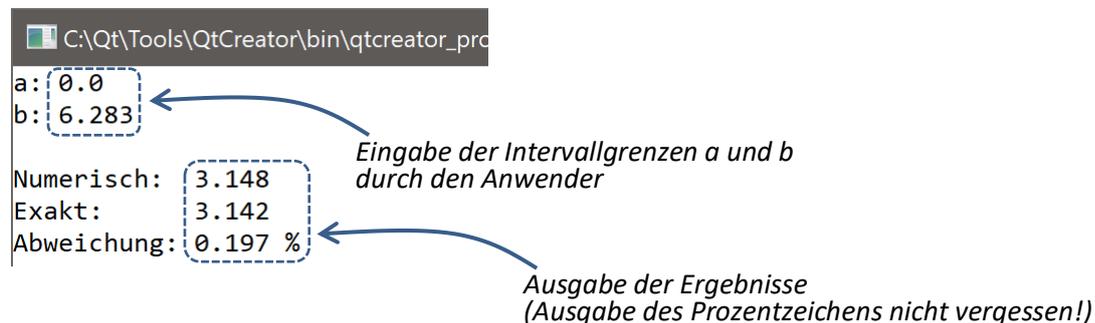
5. Ihr Programm berechnet zusätzlich zum Näherungswert f_n auch den exakten Wert des Integrals. Dies ist leicht möglich, weil die Stammfunktion von $f(x)$ bekannt ist:

$$F(x) = \frac{1}{2} \cdot (x - \sin x \cdot \cos x)$$

6. Auf dem Bildschirm werden drei Ergebnisse ausgegeben:
 - der numerisch berechnete Näherungswert f_n ,
 - der mithilfe der Stammfunktion berechnete exakte Wert,
 - die prozentuale Abweichung der beiden Werte (Ausgabe des Prozentzeichens nicht vergessen!).

Die Funktionswerte von $f(x)$ bzw. $F(x)$ werden in zwei separaten C-Funktionen berechnet, also nicht direkt im Hauptprogramm `main()`. An diese beiden Funktionen wird jeweils ein einzelner x -Wert übergeben, der daraus berechnete Wert $f(x)$ bzw. $F(x)$ wird anschließend ans Hauptprogramm `main()` zurückgegeben.

Die Bildschirmausgabe Ihres Programms soll dem folgendem Bildschirmfoto entsprechen:



Aufgabe 2: (ca. 18 Punkte)

Die drei globalen Matrizen m1, m2 und m3 sind wie folgt definiert. Gehen Sie davon aus, dass alle drei Matrizen **zunächst vollständig mit zufälligen Werten belegt** sind:

```
#define N 10
double m1[N][N], m2[N][N], m3[N][N];
```

- 2.1. Schreiben Sie eine Funktion fill1(), welche **mithilfe von geeigneten for-Schleifen** die Elemente von m1 wie folgt belegt. Verwenden Sie **keine do-while-Schleifen und keine while-Schleifen**:

```
 1.0  2.0  3.0  4.0  5.0  6.0  7.0  8.0  9.0 10.0
11.0 12.0 13.0 14.0 15.0 16.0 17.0 18.0 19.0 20.0
21.0 22.0 23.0 24.0 25.0 26.0 27.0 28.0 29.0 30.0
31.0 32.0 33.0 34.0 35.0 36.0 37.0 38.0 39.0 40.0
41.0 42.0 43.0 44.0 45.0 46.0 47.0 48.0 49.0 50.0
51.0 52.0 53.0 54.0 55.0 56.0 57.0 58.0 59.0 60.0
61.0 62.0 63.0 64.0 65.0 66.0 67.0 68.0 69.0 70.0
71.0 72.0 73.0 74.0 75.0 76.0 77.0 78.0 79.0 80.0
81.0 82.0 83.0 84.0 85.0 86.0 87.0 88.0 89.0 90.0
91.0 92.0 93.0 94.0 95.0 96.0 97.0 98.0 99.0 100.0
```

```
void fill1(void)
{
```

```
}
```

- 2.2. Erstellen Sie ein Struktogramm, welches den genauen Ablauf der Funktion fill1() zeigt:

- 2.3. Schreiben Sie eine Funktion fill2(), welche **mithilfe von geeigneten do-while-Schleifen** die Elemente von m2 wie folgt belegt. Verwenden Sie **keine while-Schleifen und keine for-Schleifen**:

```
10.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0 10.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0 10.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0 10.0  0.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0 10.0  0.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0 10.0  0.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0 10.0  0.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0 10.0  0.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 10.0  0.0
 0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0  0.0 10.0
```

```
void fill2(void)
{
```

```
}
```

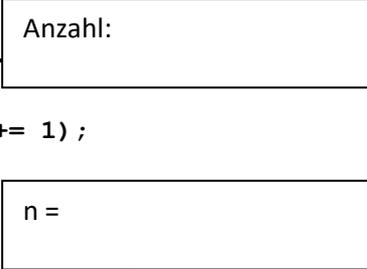
- 2.4. Erstellen Sie ein Struktogramm, welches den genauen Ablauf der Funktion fill2() zeigt:

Aufgabe 3: (ca. 7 Punkte)

Das folgende Programm sucht alle Primzahlen im Bereich von 2 bis 10000. Die Primzahlen werden auf dem Bildschirm ausgegeben, ebenso die Anzahl der gefundenen Primzahlen.

- Korrigieren Sie die fünf Fehler, die sich im Quelltext befinden!
- Wie oft wird die mit //1// markierte Zeile ausgeführt?
- Welchen Wert hat die Variable n in der mit //2// markierten Zeile?

```
#include <math.h>
int main(void)
{
    int n, t, teiler_gefunden, anzahl = 0;
    for(n = 2; n <= 10000; ++n)
    {
        teiler_gefunden = 0; //1//
        for(t = 2; t < n && teiler_gefunden == 0; t += 1);
            if(n % t = 0) teiler_gefunden = 1;
        if(teiler_gefunden == 0) {
            printf("%16d", n); anzahl = anzahl++;
        }
    }
    printf("\n%d Primzahlen gefunden\n", &anzahl); //2//
    return 0;
}
```



Aufgabe 4: (ca. 6 Punkte)

Das folgende Programm dient zur Verschlüsselung von Texten. In den mit //// markierten Zeilen werden nacheinander die Zeichenketten s1, s2, s3 an die Funktion crypto() übergeben. Fügen Sie die fehlenden Übergabeparameter zum Funktionsaufruf hinzu!

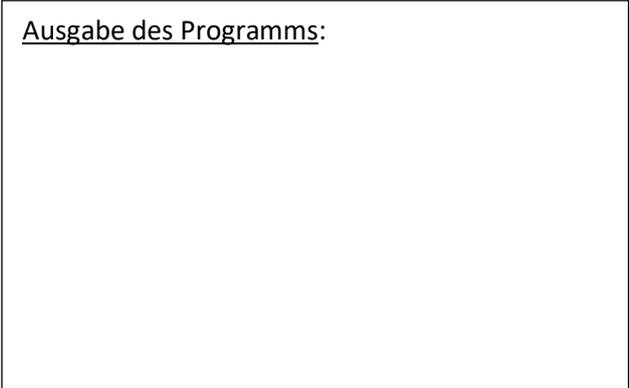
Wie sieht die Ausgabe des Programms auf dem Bildschirm aus?

```
#include <stdio.h>
#include <string.h>
void crypto(char *str);

int main()
{
    char s1[] = "Ingenieur", s2[] = "", s3[] = "x";
    crypto( ); printf("%s*\n", s1); //// Übergabe von s1
    crypto( ); printf("%s*\n", s2); //// Übergabe von s2
    crypto( ); printf("%s*\n", s3); //// Übergabe von s3
    return 0;
}

void crypto(char *str)
{
    char ch;
    int i, len = (int)strlen(str);
    for(i = 0; i < len / 2; ++i)
    {
        ch = str[i];
        str[i] = str[len - 1 - i];
        str[len - 1 - i] = ch;
    }
}
```

Ausgabe des Programms:



Aufgabe 5: (ca. 11 Punkte)

Gegeben ist die folgende 8-stellige Binärzahl: **11100110**

- 5.1. Wandeln Sie diese (vorzeichenlose) Binärzahl in eine Hexadezimalzahl um.
- 5.2. Bei der angegebenen Binärzahl handelt es sich um eine ganze Zahl **mit** Vorzeichen in 8-Bit-Zweierkomplementdarstellung. Welchen Wert hat die entsprechende Dezimalzahl?
- 5.3. Bei der angegebenen Binärzahl handelt es sich nun um eine ganze Zahl **ohne** Vorzeichen (also keine Zweierkomplementdarstellung). Welchen Wert hat die entsprechende Dezimalzahl?
- 5.4. Ein Mikrocontroller **mit 8 Bit großen Registern** addiert zur oben angegebenen Binärzahl die folgende 8-stellige Binärzahl hinzu: **01100000**
Wie lautet der Inhalt des Ergebnisregisters? Ist das Carry-Flag im Mikrocontroller gesetzt, d. h. kommt es zu einem Übertrag an der höchstwertigen Stelle der Binärzahl?
- 5.5. Ein Mikrocontroller **mit 8 Bit großen Registern** addiert zur oben angegebenen Binärzahl die folgende 8-stellige Binärzahl hinzu: **00000110**
Wie lautet der Inhalt des Ergebnisregisters? Ist das Carry-Flag im Mikrocontroller gesetzt, d. h. kommt es zu einem Übertrag an der höchstwertigen Stelle der Binärzahl?

(Platz für Nebenrechnungen)