

HOCHSCHULE
HANNOVER
UNIVERSITY OF
APPLIED SCIENCES
AND ARTS

–
Fakultät I
Elektro- und
Informationstechnik

C Programmierung

Übungsaufgaben



Aufgabe 0

Das zu entwickelnde Programm soll

- Ihren Namen, Studienrichtung und email-Adresse ausgeben
- die Summe zweier Zahlen korrekt ausgeben.
Erzeugen Sie danach durch Wahl entsprechender Datentypen einen Überlauf.
- den Kapitalertrag für ein Jahr von 180 € bei 2% Verzinsung korrekt ausgeben.
Erzeugen Sie danach durch Wahl entsprechender Datentypen einen Rechenfehler.
- die Zahl 24 durch Anwendung von Bit-Operatoren vierteln.



Aufgabe 1, 2, 3

Aufgabe 1:

- Das zu entwickelnde Programm soll die Summe von zwei ganzen Zahlen berechnen. Nach der Ausgabe der Aufgabenstellung sind die beiden Werte anzufordern. Die berechnete Summe ist zwei Zeilen tiefer auszugeben.

Aufgabe 2:

- Das zu entwickelnde Programm soll den Mittelwert von drei eingegebenen reellen Zahlen berechnen. Nach der Ausgabe der Aufgabenstellung sind die drei Zahlen anzufordern. Geben Sie den berechneten Mittelwert im Standardformat aus.

Aufgabe 3

- Dieses Programm berechnet den Zinsertrag Ihres Kapitals je nach Dauer der Anlage. Dazu sind nach Ausgabe der Aufgabenstellung das Kapital, der Zinssatz und die Dauer der Anlage in Jahren einzugeben. Die Ausgabe soll mit zwei Stellen nach dem Komma erfolgen.



Aufgabe 4, 5, 6

Aufgabe 4

- Erstellen Sie ein Programm, das die ASCII-Zeichen Nr. 48 bis 57 als Zahlenwert und als Zeichen ausgibt. Nutzen Sie hierfür eine Zählschleife.

Aufgabe 5

- Nach der Eingabe einer ganzzahligen Zahl ist zu überprüfen, ob diese Zahl < 100 ist.
- Trifft das zu, ist die Zahl um 10 zu erhöhen.
- Das ist so lange zu wiederholen, bis die Schleife verlassen werden kann.
- Überprüfen Sie den Schleifendurchlauf durch Ausgabe des aktuellen Wertes der Zahl im Schleifenkörper.
- Testen Sie Ihr Programm mit unterschiedlichen Startwerten.

Aufgabe 6

- Für die ganzzahligen Radien von $R=1$ bis $R = 10$ sollen die Kreisfläche und das Kugelvolumen mit dem Radius R berechnet und in einer Tabelle ausgegeben werden.
- Die Ausgabe soll mit zwei Stellen hinter dem Dezimalpunkt erfolgen.



Aufgabe 7 (Teil 1)

- Der Sinus eines eingegebenen Winkels ist zu berechnen. Dazu ist die folgende Reihenentwicklung heranzuziehen:

$$\sin(x) = x - x^3 / 3! + x^5 / 5! - x^7 / 7! + \dots$$

- Für die Reihenentwicklung muss der Winkel im Bogenmaß in die Gleichung eingegeben werden. Fordern Sie den Winkel in Grad an, rechnen diesen in Bogenmaß um und setzen Sie diesen Wert in die Reihenentwicklung ein. Brechen Sie die Reihe nach dem dritten Glied ab. Damit liefert die Reihe für Argumente im Bereich zwischen $-\pi/2 \leq x \leq +\pi/2$ korrekte Ergebnisse.



Aufgabe 7 (Teil 2)

- Strukturieren Sie Ihr Programm mit Unterprogrammen. Erstellen Sie hierfür die folgenden Funktionen:
 - `float winkelBogen(float)`: Umrechnung eines Winkels von Grad in Rad (Bogenmaß)
 - `int fakultaet(int)`:
Berechnung der Fakultät einer ganzen Zahl
 - `float potenz(float, int)`:
Berechnung der ganzzahligen Potenz einer reellen Zahl

Rufen Sie diese Funktionen an geeigneter Stelle in Ihrem Hauptprogramm auf, um den Sinus eines Winkel mit einer Taylor-Reihe zu berechnen.
- Die Reihe liefert für Argumente im Bereich zwischen $-\pi/2 \leq x \leq +\pi/2$ hinreichend genaue Ergebnisse. Wie müssten Sie Ihr Programm erweitern, dass die Reihe für beliebige Argumente richtige Ergebnisse liefert?



Aufgabe 8 (Teil 1)

- Erstellen Sie ein Feld von 10 int-Werten.
Initialisieren Sie das Feld mit beliebigen Daten.
- Berechnen Sie die Summe aller Feldelemente und geben Sie diese Summe aus.
- Berechnen Sie den Mittelwert aller Feldelemente und geben Sie den Mittelwert aus.
- Ermitteln Sie das größte Feldelement und geben Sie dieses Feldelement und seine Position (Index) im Feld aus.
- Erstellen Sie vor der Programmerstellung ein Struktogramm.



Aufgabe 8 (Teil 2)

- Strukturieren Sie Ihr Programm mit folgenden Unterprogrammen
 - `int summe(int [10]):`
Berechnung der Summe aller Elemente des übergebenen Feldes `int [10]`
 - `float mittelwert(int [10]):`
Berechnung des Mittelwerts aller Elemente des übergebenen Feldes `int [10]`
 - `int maximum(int [10]):`
Suche nach dem größten Feldelement und Ausgabe seiner Position (Index) im Feld.
- Passen Sie Ihr Struktogramm entsprechend an.



Aufgabe 8 (Teil 3)

- Erweitern Sie Ihr Hauptprogramm `int main ()`, damit der Benutzer ein beliebiges 10er Feld (siehe Aufgabe 8, Teil 1) über die Tastatur eingeben kann. Schreiben Sie dazu eine Funktion:

```
void feldEinlesen(int[10]);
```

- Entwickeln Sie ein Unterprogramm, um ein 10er Feld am Bildschirm anzeigen zu können.

```
void feldAusgeben(int[10]): Ausgabe eines 10er Feldes am  
Bildschirm
```

- Berechnen Sie für das vom Benutzer eingegebene Feld den Mittelwert, die Summe, das größte Element und geben Sie das Feld aus.
(siehe Aufgabe 8, Teil 1 und Teil 2) .
- Erstellen Sie für das so erweiterte Programm ein Struktogramm.



Aufgabe 9 (Teil 1)

- Erstellen und testen Sie ein Programm, das für eine beliebige positive Ganzzahl die nächsthöhere gerade Zahl ausgibt. Falls die Zahl bereits gerade ist, ist die Zahl selbst auszugeben. Die Zahl ist durch den Benutzer einzugeben.
- Erstellen Sie hierzu folgende Unterprogramme.
 - `void eingeben(int*)`: erhält einen Zeiger auf eine Zahl und ermöglicht die Eingabe dieser Zahl durch die Tastatur
 - `void korrigieren(int*)`: erhält einen Zeiger auf die vorher eingegebene Zahl und korrigiert diese, falls sie ungerade ist.
 - `void ausgeben(int*)`: erhält einen Zeiger auf die vorher eingegebene und korrigierte Zahl und gibt diese mit Wert und Adresse am Bildschirm aus.
- Erstellen Sie das Struktogramm für Ihr Hauptprogramm und für die Unterfunktionen.
- Testen Sie Ihr Programm mit unterschiedlichen Testfällen.



Aufgabe 9 (Teil 2)

- Erweitern Sie Ihr Programm um eine weitere Funktion, die überprüft, ob die eingegebene Zahl mindestens 5-stellig ist und diese bei Bedarf korrigiert. Füllen Sie dazu die fehlenden Stellen rechts neben der Zahl mit 0 auf.
 - `void korrigieren2(int*)`: erhält einen Zeiger auf eine vorher eingegebene Zahl und erweitert diese bei Bedarf auf 5 Stellen.
- Ändern Sie die Eingaberoutine so ab, dass der Anwender darauf hingewiesen wird, dass er keine 0 eingeben darf.
- Erstellen Sie das Struktogramm für die Funktion `korrigieren2`.
- Testen Sie Ihre neue Funktion mit unterschiedlichen Testfällen.



Aufgabe 9 (Teil 3)

- Erstellen Sie eine Schleife, die die Eingabe, Ausgabe und die Korrekturen solange wiederholt, bis die Zahl 0 eingegeben wird. Weisen Sie den Anwender auf dieses Abbruchkriterium hin.
Das Programm wird nach Eingabe dieser Zahl beendet.
- Erstellen Sie das Struktogramm Ihres so modifizierten Programms.



Aufgabe 10

Mit Ihrem Programm wird eine Zeichenkette von maximal 50 Zeichen eingelesen und in einer passenden Datenstruktur abgelegt. Die einzelnen Teilaufgaben können in Unterprogrammen umgesetzt werden. Gehen Sie davon aus, dass nur ASCII Zeichen ohne Umlaute eingegeben werden. Erweitern Sie in einem zweiten Schritt Ihr Programm um die Erkennung von Umlauten

- Geben Sie die Zeichenkette wieder aus. Ermitteln Sie die Anzahl der Zeichen innerhalb der Zeichenkette.
- Geben Sie die Zeichenkette in umgekehrter Reihenfolge wieder aus
- Ermitteln Sie die Anzahl der Großbuchstaben
- Ermitteln Sie die Anzahl der Vokale in der Zeichenkette und geben Sie die Anzahl aus.
- Geben Sie die Zeichenkette nur mit Konsonanten aus, Vokale sind ersatzlos gestrichen. Geben Sie die Länge der so entstandenen Zeichenkette und die prozentuale Kompressionsrate (Textlänge ohne Vokale bezogen auf Länge mit Vokalen) aus.

Erstellen Sie für Ihre Lösungen zu jeder Teilaufgabe ein Struktogramm.



Aufgabe 11

Ihr Programm soll analysieren, welche die ersten 5 Worte eines Satzes (max. 50 Zeichen inkl. Leerzeichen) sind. Lösen Sie dazu folgende Teilaufgaben.

- Lesen Sie den Satz als Zeichenkette ein und legen Sie ihn in einer passenden Datenstruktur ab.
- Legen Sie ein `char*`-Feld für 5 Wörter mit dem Namen `wortSpeicher` an und reservieren Sie mit dem Befehl `calloc` Speicher für mindestens 20 Zeichen pro Wort.
- Ermitteln Sie die ersten 5 Worte innerhalb der eingegebenen Zeichenkette und kopieren Sie jedes einzelne Wort in den Wortspeicher.
Hinweis zur erlaubten Eingabe eines Satzes: Worte sind durch das Leerzeichen getrennt, der Satz endet immer mit einem „.“
- Geben Sie jedes Wort des Wortspeichers auf dem Bildschirm und in einer Textdatei aus. Geben Sie die Anzahl der Wörter aus, falls der Satz weniger als 5 Wörter hat.

Erstellen Sie für Ihre Lösung ein Struktogramm.



Aufgabe 12 (Teil 1)

In einem Lager eines Automobilherstellers werden Fahrzeuge zwischengelagert. Ein Lagerplatz wird beschrieben durch eine Struktur `PLATZ`, die folgende Komponenten enthält:

- `nr` : Seriennummer des Fahrzeugs, positive ganze achtstellige Zahl
- `x` : x-Koordinate des Platzes, positive ganze Zahl zwischen 1 und 10
- `y` : y-Koordinate des Platzes, positive ganze Zahl zwischen 1 und 10
- `typ` : Zeichenkette mit Fahrzeugtyp ("Kombi", "Limousine", "Cabrio")

Finden Sie die passende Datenstruktur für `PLATZ` und deklarieren Sie ein Feld mit Namen `lager` mit 20 Einträgen von Typ `PLATZ` .

Initialisieren Sie in Ihrem Hauptprogramm die ersten 5 Einträge des Feldes `lager` mit beliebigen Werten.

Geben Sie in einer Schleife die Seriennummer und den Typ der ersten 5 Einträge am Bildschirm aus.



Aufgabe 12 (Teil 2)

Erstellen Sie nun ein Programm zur **dynamischen Lagerverwaltung**, mit dem Sie an beliebigen Stellen Fahrzeuge einlagern und auslagern können. Gehen Sie hierzu folgendermaßen vor.

- Erstellen Sie ein neues Hauptprogramm. Initialisieren Sie alle Einträge Ihres Feldes `lager` mit folgenden *default*-Werten: `nr = 0`, `x` und `y = 0`, `typ = "--"`. Mit diesen Werten werden freie Lagerplätze gekennzeichnet!
- Schreiben Sie eine Funktion `ausgabe`, die den kompletten Lagerbestand ausgibt mit Seriennummer, Position und Fahrzeugtyp.
- Schreiben Sie eine Funktion `einlagern`, die an der ersten freien Stelle Ihres Feldes `lager` die übergebenen Werte schreibt. Überprüfen Sie in Ihrer Funktion, ob der vorgegebene Lagerplatz frei oder bereits Erstellen Sie vor der Programmierung ein Struktogramm.
- Schreiben Sie eine Funktion `auslagern`, die ein Fahrzeug mit der angegebenen Seriennummer aus dem Feld entfernt, indem die Werte der Struktur auf die *default*-Werte zurückgesetzt werden.
- Schreiben Sie eine Funktion `inventur`, die ermittelt, wieviele Fahrzeuge insgesamt und wieviele Kombis, Limousinen und Cabrios im Lager sind.



Ende der Übungsaufgaben

