

Ingenieurinformatik (FK 03)
Übung 2

ÜBUNG 2A (SCANF UND PRINTF)

Einführung

Die C-Bibliotheksfunktionen **scanf()** und **printf()** dienen zur formatierten Ein- bzw. Ausgabe.

scanf() wird verwendet, um ASCII-Tastatureingabe als - im Formatspezifizierer angegeben Datentypen zu interpretieren - und die Eingabe in Variablen zur Weiterverarbeitung abzulegen. **printf()** wandelt den Inhalt einer Variablen in eine – dem Formatspezifizierer entsprechende – Bildschirmausgabe um.

Folgende Tabelle stellt einige zu testenden Formatspezifizierer dar:

Datentypen	Formatspezifizierer	Bemerkung
int, unsigned	%d <i>oder</i> %i	vorzeichenbehaftet
int, unsigned	%u	vorzeichenlos
unsigned, int	%x <i>oder</i> %X	vorzeichenlos, hexadezimal
long int	%ld	vorzeichenbehaftet
double	%lf	nur für scanf()
double	%f	für printf()
float	%f	
float, double	%e <i>oder</i> %E	Exponentialdarstellung
char *	%s	Zeichenkette

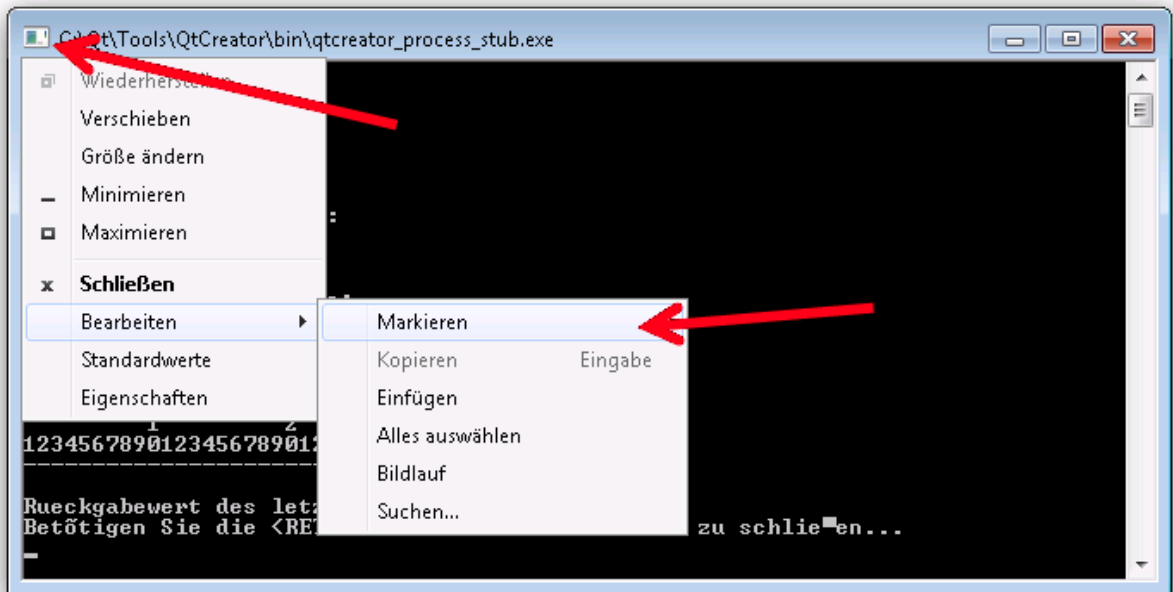
Ziel

Es sollen ein C-Programm erstellt werden, dass die unterschiedlichen Formatspezifizierer testet und ermittelt welche Rückgabewerte die Funktionen liefern.

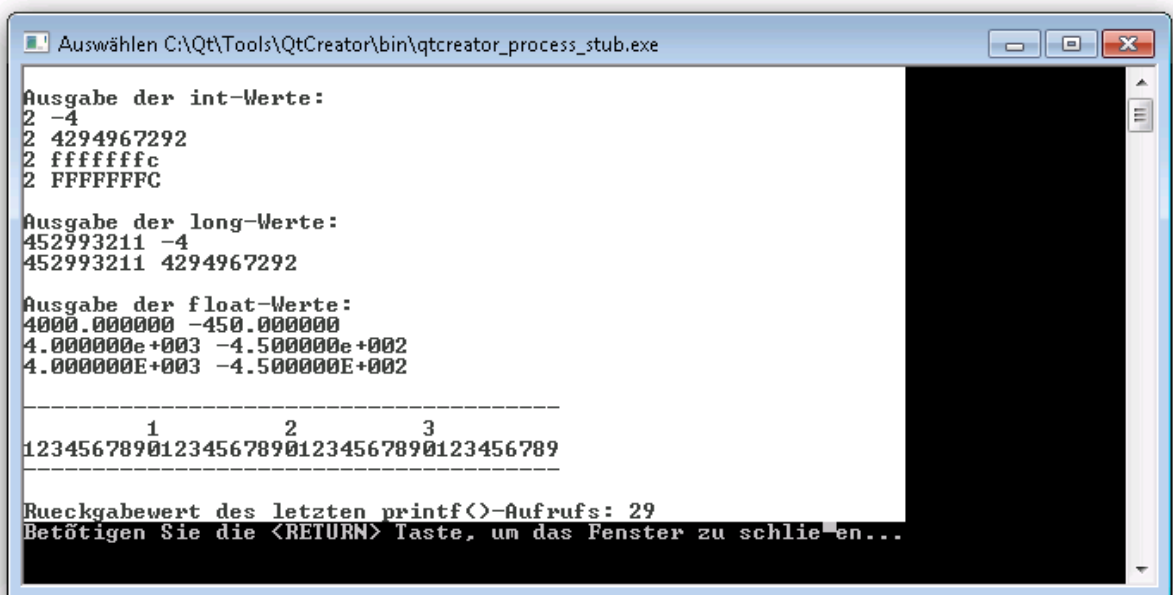
Durchführung

- Starten Sie das Programm **QT Creator (Community)**.
- Wählen Sie als Arbeitsbereich (Workspace) den Pfad **U:\workspace**
- Erstellen Sie in diesem Arbeitsbereich ein **C-Projekt** mit dem Namen **ein_ausgabe** gemäß der Anleitung auf:
http://pc01-lsw.ee.hm.edu/wicked/QTC_Projekterstellung
- Kopieren Sie nun die bereitgestellte Datei **ein_aus.c** in das Verzeichnis **U:\workspace\ein_ausgabe** mithilfe des Windows-Explorer.
- Fügen Sie nun über **QT Creator** die bestehende Quelldatei **ein_aus.c** hinzu.
- Da diese Datei bereits eine **main**-Funktion besitzt und nur eine **main**-Funktion in einem Projekt erlaubt ist, muss die erstellte **main.c** aus dem Projekt entfernt werden. Klicken Sie dazu auf den Dateinamen **main.c** und drücken die **<Entf>**-Taste.
- Erstellen Sie das bereitgestellte Programm.

- Starten Sie das Programm, markieren die Ausgabe (s. nachfolgende Screenshots) und speichern Sie die Ausgabe zum späteren Vergleich in einer Textdatei mithilfe des Programms **TextPad**.



Nach dem Ziehen der Maus über den zu kopierenden Bereich drücken Sie **<Enter>**, um den Bereich in die Zwischenablage zu kopieren.



Mit **<Strg>-V** im **TextPad**-Editor erhalten Sie dann die Kopie in der Textdatei.

- **Frage A1:** Worin besteht der Unterschied zwischen den Formatspezifizieren **%x** und **%X** bzw. **%e** und **%E**?
- Ändern Sie die Formatspezifizier **aller printf ()**-Aufrufe so ab, dass die Zahl 15 zwischen Prozentzeichen und Konvertierzeichen (z. B. "**%15d %15d**") steht.
- **Frage A2:** Welche Änderung können Sie in der Ausgabe feststellen und wie verhält sich der Rückgabewert der letzten **printf ()**-Anweisung?

- Ändern Sie die Formatspezifizier **aller Fließkommazahlengaben** so ab, dass die Zahl **15.2** zwischen Prozentzeichen und Konvertierzeichen (z. B. "%15.2f %15.2f") steht.
- **Frage A3:** Welche Änderung können Sie in der Ausgabe feststellen und wie verhält sich der Rückgabewert der letzten **printf()**-Anweisung?
- Ändern Sie die Formatspezifizier **aller Ganzzahlangaben** so ab, dass die Zahl **-15** zwischen Prozentzeichen und Konvertierzeichen (z. B. "%-15u %-15u") steht.
- **Frage A4:** Welche Änderung können Sie in der Ausgabe feststellen und wie verhält sich der Rückgabewert der letzten **printf()**-Anweisung?
- **Frage A5:** Was gibt der Rückgabewert (In **ein_aus.c** → Inhalt der Variablen: **rueckgabe**) einer **printf()**-Anweisung an?

- Entfernen Sie den Blockkommentar und aktivieren Sie damit die Eingabe der **int**-Werte und die entsprechende nachfolgende Ausgabe.
- Starten Sie das erstellte Programm über **QT Creator**.
- Geben Sie zuerst die Werte **-5** und **4** ein.
Wiederholen Sie die Ausführung und geben nun **-5** und **xyz** ein.
Starten Sie Programm erneut und geben und zuerst **xyz** ein.
- **Frage A6:** Wie verhalten sich die Variablenwerte von **wert1** und **wert2**? Was können Sie über den Rückgabewert (Aktualisierte Inhalt der Variablen: **rueckgabe**) der **scanf()**-Funktion sagen?

- **Zusatzübung für zuhause**
Erweitern Sie das Programm um die Eingabe der zwei **double**-Werte in den Variablen **dwert1** und **dwert2** und geben die eingelesenen Variableninhalte nachfolgend aus.
- Testen Sie Ihr Programm.

ÜBUNG 2B (STRUKTOGRAMM UND REALISIERUNG)

Einführung

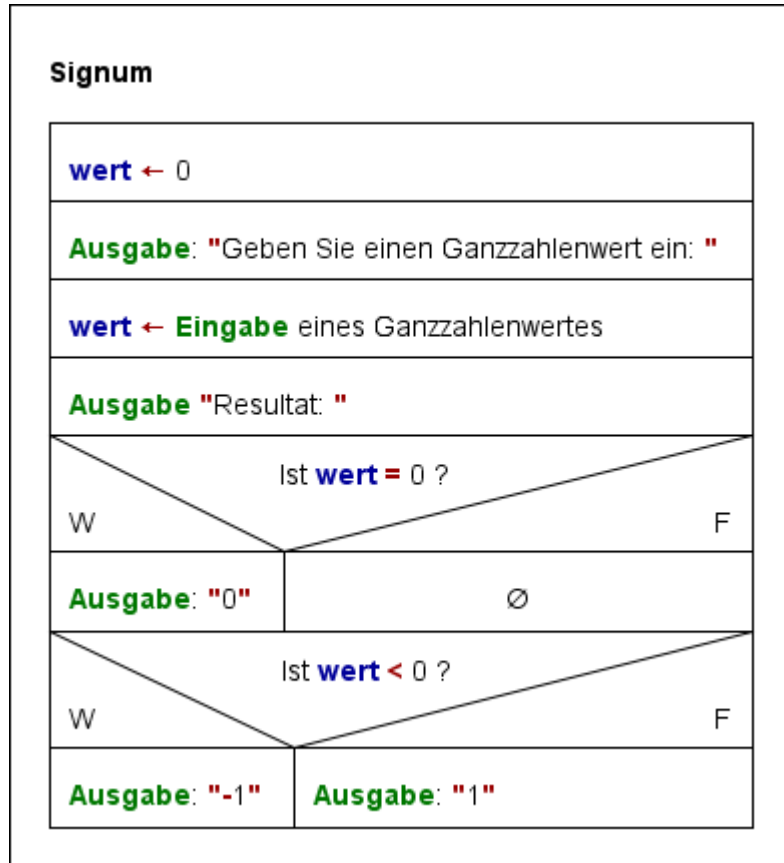
Die Signumfunktion überprüft, ob eine Zahl negativ, positiv oder 0 ist.

Für die Eingabe einer negativen Zahl soll der Wert **-1**, für positive der Wert **1** und für die Zahl **0** soll **0** ausgegeben werden.

Ziel

Es sollen ein C-Programm mithilfe der Vorgabe eines Struktogramms erstellt werden, Fehler identifiziert und sowohl im Programm als auch im Struktogramm korrigiert werden.

Vorgabe



Durchführung

- Schließen Sie ggf. bereits geöffnete Projekte (über den Menüpunkt *Datei* → *Alle Projekte und Editoren schließen*) und erstellen ein neues **C-Projekt** mit dem Namen **signum**.
- Realisieren Sie die Vorgaben des Struktogramms in der Funktion **main()**.
- Erstellen Sie das Programm und testen Ihre Realisierung für unterschiedliche Werte.
Frage B1: Für welche(n) Wert(e) funktioniert das Programm nicht richtig?
- Korrigieren Sie das Struktogramm und realisieren Sie die Korrektur im C-Quellcode.
- **Zusatzübung für zuhause**
 Sollte keine Zahl eingegeben werden, dann soll das Programm mit der Fehlermeldung **Fehler: Keine Zahl eingegeben!!** enden ohne das Resultat auszugeben.
 Ändern Sie das Struktogramm entsprechend ab und realisieren Sie die Erweiterung im C-Quellcode.

ZUSATZÜBUNG 2C (PREISBERECHNUNG EINES MIETWAGENS)

Ziel

Es soll ein C-Programm zur Berechnung eines Mietpreises erstellt werden.

Der Benutzer soll dazu folgende Eingaben machen:

- Wagenkategorie, (1) klein oder (2) groß
- Mietzeit (Anzahl der ganzen Tage)
- Gefahrenen ganze Kilometer

Berechnungsalgorithmus:

Ein kleiner Wagen hat 60 km frei und kostet 60 Euro pro Tag; jeder zusätzlich gefahrene Kilometer kostet 0.40 Euro.

Große Wagen haben 90 km frei, kosten 90 Euro je Tag; jeder zusätzlich gefahrene Kilometer 0.60 Euro.

Falscheingaben müssen nicht abgefangen werden.

Durchführung

- Erstellen Sie ein Struktogramm mithilfe der Applikation **Structorizer**.
s. <http://structorizer.fisch.lu/>
- Schließen Sie ggf. bereits geöffnete Projekte (über den Menüpunkt *Datei* → *Alle Projekte und Editoren schließen*) und erstellen ein neues **C-Projekt** mit dem Namen **preisberechnung**.
- Realisieren Sie die Vorgaben des Struktogramms in der Funktion **main()**.
- Erstellen Sie das Programm.
- Testen Sie Ihr Programm, korrigieren Sie ggf. Ihr Struktogramm und Ihren Quellcode.
Warnungen und Fehler müssen zur Abnahme behoben werden.
- Erweitern Sie das Programm, so dass die Eingabe und Berechnung solange wiederholt wird wie für die Wagenkategorie ein gültiger Zahlenwert eingegeben wird.
Ändern Sie das Struktogramm entsprechend ab und realisieren Sie die Erweiterung im C-Quellcode.