

Der C-Coding Styleguide ist einzuhalten!

Folgende Einstellungen sind für Debug und Release (All Configurations) vorzunehmen:

Einstellung	Wert
Solution Platform	x86
Properties->Conf. Properties->C/C++->General->Warning Level	Level4 (/W4)
Properties->Conf. Properties->C/C++->General->Treat Warnings As Errors	Yes (/WX)
Properties->Conf. Properties->C/C++->General->SDL checks	Yes (/sdl)
Properties->Conf. Properties->C/C++->Code Generation->Basic Runtime Checks	Default
Properties->Conf. Properties->C/C++->Code Generation->Security Checks	Enable Security Checks (/GS)

Solution muss in Debug und Release fehlerfrei kompilierbar sein.

Packen Sie das gesamte Verzeichnis der Solution in eine Zip-Datei und laden Sie diese in Moodle pünktlich hoch.

Aufgabe 1:

Implementieren Sie ein Programm, welches ein vorzeichenloses Integer-Array bestehend aus 10 Elementen definiert. Initialisieren Sie dieses Array aufsteigend mit den Werten 0U-9U.

Definieren Sie zwei Zeiger, welche auf Elemente des Arrays zeigen können. Implementieren Sie die folgenden Operationen mit Zeigern:

- Zuweisung mit NULL
- Zuweisung einer Adresse (Verwenden Sie hierzu Arrayelemente)
- Inkrementieren und Dekrementieren (mit anschließendem printf())
- Subtrahieren und Addieren (mit anschließendem printf())
- Differenz zweier Zeiger (Ausgabe mit printf())
- Vergleichsoperatoren

Fügen Sie im Code genügend Kommentare ein.

Aufgabe 2:

Implementieren Sie ein Programm, welches ein Integer-Array bestehend aus 5 Elementen definiert. Initialisieren Sie dieses Array mit den Werten 1, 2, 3, 4 und 5 bei der Definition:

```
int aiTestArray[5] = {1, 2, 3, 4, 5};
```

Ein Integer-Zeiger piValInArray soll nun auf das dritte Element (Wert 3, Index 2) zeigen. Drucken Sie mit printf() den Wert über den dereferenzierten Zeiger aus. Inkrementieren Sie den Zeiger dreimal und drucken Sie erneut den Wert über den dereferenzierten Zeiger aus!

- Was wird jetzt ausgegeben?

Schreiben Sie die Antwort vor main als Kommentar.

Aufgabe 3:

Implementieren Sie ein Programm, welches ein Integer-Array mit 100 Elementen definiert. Implementieren Sie eine Funktion welche dieses Array mit den Werten 50, 49, ..., 0, ... -48, -49 initialisiert. Implementieren Sie eine weitere Funktion, welche das Array auf dem Bildschirm ausgibt. Dabei sollen jeweils 10 Elemente in einer Zeile ausgegeben werden. Verwenden Sie hierzu auch das Steuerzeichen '\t'.

Kopieren Sie die drei BubbleSort-Funktionen aus dem Skript in Ihr Projekt und machen Sie sich mit der Funktionsweise dieser Funktionen vertraut. Testen Sie die drei BubbleSort-Funktionen unter Nutzung Ihrer beiden Funktionen.

- Welcher Nutzen hat das „-uiY“ in der inneren Schleife?

Schreiben Sie die Antwort vor main als Kommentar.

Aufgabe 4:

Implementieren Sie ein Programm, welches ein Array von sieben Elementen des Datentyps `sAddress_t` definiert und initialisiert (siehe Skript). Definieren Sie einen passenden Zeiger `psAddress`, der auf das erste Element zeigen soll. In einer `for`-Schleife soll `psAddress` solange inkrementiert werden, bis das Ende des Arrays erreicht ist. Vor der Inkrementierung soll der Wert des Schleifenzählers (`unsigned int`) zum `uiZipCode` des Elementes hinzuaddiert werden, auf welches `psAddress` zeigt.

Geben Sie anschließend den `uiZipCode` von allen Elementen des Arrays auf dem Bildschirm aus.

Setzen Sie nun den Zeiger auf das Element des Arrays mit dem Index 2. Addieren Sie den Wert 2 zum Zeiger `psAddress` und geben Sie unter Nutzung des Zeigers das entsprechende Element des Arrays auf dem Bildschirm aus.

Addieren Sie anschließend den Wert 3 zum Zeiger `psAddress` und geben Sie unter Nutzung des Zeigers das entsprechende Element des Arrays auf dem Bildschirm aus.

- Was wird jetzt ausgegeben?

Schreiben Sie die Antwort als Kommentar über `main`.

Aufgabe 5:

Implementieren Sie ein Programm, welches **dynamisch (malloc)** Speicher für 100 Zahlenwerte vom Datentyp `short int` allokiert. Verwenden Sie einen zweiten Zeiger, der ebenfalls auf den Beginn des allokierten Speichers zeigt. Diesen zweiten Zeiger verwenden Sie, um Werte in das Array zu schreiben (von 100 bis 1). Gehen Sie dabei wie folgt vor (`for`-Schleife): Zuweisen, Inkrementieren, Zuweisen, Inkrementieren... Geben Sie danach den gesamten Inhalt des Arrays auf den Bildschirm aus. Sortieren Sie danach dieses Array mit `BubbleSort` und drucken Sie anschließend den gesamten Inhalt des allokierten Speichers wieder auf dem Bildschirm aus.

- Wie kann die Laufzeit von `BubbleSort` verbessert werden, falls ein Array schon weitestgehend vorsortiert ist (z.B. -5, -4, -3, -1, -2, 0, 1, 2, 3, 6, 5, 7)?

Hinterlegen Sie die Antwort als Kommentar vor `main`.