Übungen Ing.-Inf. – KE 5

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 eine Enumeration für die Studiengänge der Fakultät EMI (mindestens 5) deklariert. Deklarieren Sie danach ein typedef für diese Enumeration. Definieren Sie in main eine Variable von diesem „neuen“ Datentyp und weisen Sie dieser einen gültigen Wert (Element der Enumeration) zu. Implementieren Sie danach eine mehrfache Selektion (switch) mit dieser Variablen und geben Sie im entsprechenden case den vollständigen Namen des Studienganges auf der Konsole aus. Wie viele Bytes benötigt die Variable von diesem neuen Datentyp? Geben Sie mittels sizeof diesen Wert auf der Konsole aus.

Aufgabe 2:
Implementieren Sie ein Programm, welches ein lokales char-Array acFirstName mit 20 Elementen definiert, ohne unmittelbar eine Wertzuweisung durchzuführen (d.h. Variablendefinition ohne Initialisierung). Weisen Sie die Buchstaben Ihres Vornamens einzeln den Arrayelementen zu und geben Sie das char-Array auf dem Bildschirm aus.

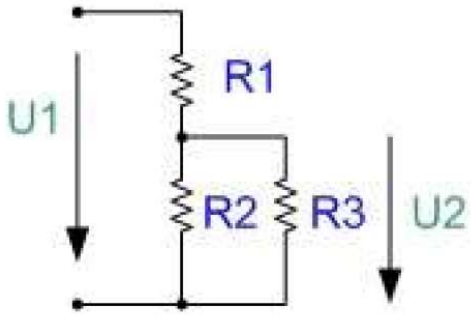
```
//Example
acFirstName[0] = 'H';
acFirstName[1] = 'a';
acFirstName[2] = 'n';
acFirstName[3] = 's';

printf("%s", acFirstName);
```

- Was fällt Ihnen dabei auf?
- Wann beendet die printf-Funktion die Ausgabe der Zeichen auf der Konsole?

Aufgabe 3:
Implementieren Sie ein Programm, welches ein lokales Integer-Array aiArray mit 10 Elementen definiert, ohne unmittelbar eine Wertzuweisung durchzuführen (d.h. Variablendefinition ohne Initialisierung). Beschreiben Sie anschließend das Array mittels einer Schleife. Es soll jedem Element der Wert 42 zugewiesen werden. Geben Sie anschließend alle Elemente des Arrays wieder auf dem Bildschirm aus.
Sollte nach einem zusätzlichen
`printf("%d", aiArray[10]);`
der Wert 42 auf der Konsole erscheinen, haben Sie Arrays in C höchstwahrscheinlich **nicht** verstanden!

Übungen Ing.-Inf. – KE 5

Aufgabe 4:
Implementieren Sie ein Programm, welches die Spannung U2 in Abhängigkeit von R3 berechnet (siehe Schaltung) und in einem Array speichert (U2 = f(R3)). Berechnen Sie U2 für die folgenden Werte von R3 [Ohm]: 0, 10, 100, 1k, 10k, 100k, 1M. U2 und R3 sind dabei Arrays.
Wie viele Elemente müssen diese Arrays enthalten?
Welcher Datentyp müssen diese Arrays haben?
Berechnen Sie in der Schleife erst einmal die Parallelschaltung R2||R3 mit einer geeigneten Gleichung.
Setzen Sie danach die Spannungsteilergleichung an.
Geben Sie anschließend die Werte des Array U2 in Abhängigkeit von R3 aus. Es gilt: R1 = R2 = 100 Ohm, U1 = 10 V. R1, R2 und U1 sollen konstante Variablen sein.



Aufgabe 5:
Implementieren Sie ein Programm, welches eine Struktur mit dem Namen Bar deklariert (siehe unten). Definieren Sie in main ein Array mit fünf Elementen von struct Bar. Initialisieren Sie dieses Array bei der Definition mit Ihren fünf Stammbars/-kneipen/-gaststätten/-clubs. Geben Sie mit einer For-Schleife alle fünf Elemente auf der Konsole aus. Ein Array-Element soll dabei in einer Zeile ausgegeben werden.

```
struct Bar
{
    char acBarName[30];
    char acSpecialOffer[30];
    char acTown[20];
}
```

Aufgabe 6:
Implementieren Sie ein Programm, welches eine Struktur mit dem Namen Employee und den Elementen für Nachname, Vorname und Alter deklariert. Überlegen Sie sich sinnvolle Datentypen, Arraygrößen und **englische** Variablennamen (siehe Programmierrichtlinien) für die Elemente. Definieren Sie in *main* ein Array mit drei Elementen dieser Struktur. Weisen Sie in einer Schleife diesen Elementen Werte mit der Funktion scanf_s zu und drucken Sie diese anschließend wieder mit einer Schleife auf dem Bildschirm aus. Geben Sie als letzte Ausgabe die Größe des komplexen Datentyps struct Employee auf dem Bildschirm aus. Vorname und Nachname **ohne** Leerzeichen eingeben! Keine zu langen Zeichenketten eingeben! Maximale Anzahl Zeichen: Länge der Zeichenkette - 1

Aufgabe 7:
Kopieren Sie die den Inhalt der c-Datei aus Aufgabe 6 in Main.c des neuen Projektes und erweitern Sie dann das Programm:
Deklarieren Sie eine neue Struktur mit dem Namen Address. Diese Struktur enthält als Elemente den Stadtnamen, Postleitzahl und Straßennamen. Überlegen Sie sich sinnvolle Datentypen und englische Variablennamen für die Elemente. Die Struktur Address soll nun wiederum Element der Struktur Employee sein. Weisen Sie in einer Schleife den drei Elementen des Arrays (Datentyp struct Employee) jetzt auch jeweils Werte für die innere Struktur Address mit der Funktion scanf_s zu. Drucken Sie anschließend das Array von Strukturen Employee mit einer Schleife auf dem Bildschirm aus. Geben Sie als letzte Ausgabe die Größe der komplexen Datentypen struct Address und struct Employee auf dem Bildschirm aus.
Vorname, Nachname, Stadt und Straßennamen ohne Leerzeichen eingeben. Keine zu langen Zeichenketten eingeben! Maximale Anzahl Zeichen: Länge der Zeichenkette - 1