

Klausur Programmieren 1

HAW-Hamburg, Fakultät Technik und Informatik, Department Informations- und Elektrotechnik
Prof. Dr. Robert Heß, 24.1.2013, Bearbeitungsdauer: 180 Min.

Erlaubte Hilfsmittel: Vorlesungsunterlagen, Lösungen aus dem Praktikum und C/C++ Einführungsbücher.

Ergebnis: von 90 Punkten

Note: Punkte.

1 Einleitung

Es soll ein Kopfrechnen-Trainer erstellt werden. Der Benutzer muss der Reihe nach zehn Multiplikations-Aufgaben lösen. Danach wird ihm die Erfolgsquote in Prozent ausgegeben.

2 Programmieraufgaben

Aufgabe 1 (5 Punkte)

Legen Sie im Hauptprogramm einen zweidimensionalen Vektor für zehn Aufgaben mit jeweils zwei Faktoren und dem Ergebnis an.

Aufgabe 2 (15 Punkte)

Erstellen Sie eine Funktion mit Namen `AufgabenErstellen(...)`, die den Vektor mit zufälligen Aufgaben (ohne Lösungen) füllt. Verwenden Sie dafür die Funktion `rand()` aus der Header-Datei `stdlib.h`, die eine ganze Zahl zwischen 0 und `RAND_MAX` zurückgibt. (Hinweis: mit dem Modulo-Operator können Sie den Zahlenbereich von `rand()` einschränken.) Die Faktoren für die Aufgaben sollen zwischen 2 und 10 liegen. Hinweis: Damit der Benutzer nicht immer die gleichen Aufgaben gestellt bekommt sollten Sie die Header-Datei `time.h` einbinden und zu Beginn die folgende Zeile einmal aufrufen: `srand((unsigned)time(NULL));`

Aufgabe 3 (15 Punkte)

Erstellen Sie eine Funktion mit Namen `ErgebnisseAbfragen(...)`, die dem Benutzer der Reihe nach die zehn Aufgaben stellt und die Ergebnisse abfragt. (Hinweis: Sie können hier die Funktion `GetInt(...)` aus dem Praktikum verwenden.) Die Ergebnisse werden in den zuvor erstellten Vektor eingetragen.

Aufgabe 4 (10 Punkte)

Erstellen Sie eine Funktion mit Namen `AufgabenPruefen(...)`, welche die Ergebnisse prüft, und die Anzahl der richtigen Ergebnisse zurückgibt.

Aufgabe 5 (15 Punkte)

Fügen Sie die Programmstücke zu einem lauffähigen Programm zusammen:



Achten Sie auf einen guten Programmierstil (Vermeidung globaler Variablen, sinnvolle Variablennamen, Quellcode einrücken und kommentieren, keine absoluten Sprünge mit `goto` etc.).

3 Verständnisfragen

Aufgabe 6 (12 Punkte)

Was ist der Zahlenbereich von <i>signed char</i> ?	
Was ist 30_{16} dezimal?	
Was ist 14_{10} dual?	
Was ist 110011010111_2 hexadezimal?	
Definieren Sie einen Vektor mit zehn Elementen vom Typ <i>unsigned</i>	
Definieren Sie eine Zeichenkette für zehn Buchstaben	

Aufgabe 7 (8 Punkte)

Was ergeben folgende Ausdrücke?

$5/4*3$	
$5*4\%3$	
$5\%4*3$	
$4<3 3!=4$	

$3\&7$	
$3\^4$	
$1<<5$	
$3\%2?"A": "B"$	

Aufgabe 8 (6 Punkte)

Betrachten Sie für die Variable *Zahl* vom Typ *unsigned* folgende Programmzeile:

```
printf("%d\n", Zahl/3*3);
```

Was wird auf dem Bildschirm ausgegeben? Begründen Sie Ihre Aussage.

Aufgabe 9 (4 Punkte)

Extrahieren Sie aus einer Variable *Wert* vom Typ *unsigned* die niedrigwertigste oktale Ziffer.

```
printf("%d", _____);
```