

Klausur PR2

HAW-Hamburg, Fakultät Technik und Informatik, Department Informations- und Elektrotechnik

Dr. Robert Heß, 6.7.2007

Bearbeitungsdauer: 90 min

Hilfsmittel: Vorlesungsunterlagen und C/C++ Einführungsbücher (z.B. Erlenkötter)

Jegliche Art von elektronischen Hilfsmitteln sind nicht erlaubt.

1. Aufgabe (6 Punkte)

Folgende Variablen sind definiert:

```
int Zahl[] = { 1, 1, 2, 3, 5, 8 };  
char *Text = "Guten Tag";
```

Was ergeben folgende Ausdrücke?

```
*Zahl      _____  
*Text      _____  
Zahl[3]    _____  
Text[2]    _____  
*(Zahl+4)  _____  
*(Text+7)  _____
```

2. Aufgabe (6 Punkte)

Für eine verkettete Liste wurde folgende Struktur erstellt

```
struct sElement {  
    struct sElement *prev;  
    struct sElement *next;  
    int Zahl;  
};
```

Aus einer Kette soll ein Element entfernt werden, für das mit `malloc()` Speicher reserviert wurde. Die Variablen `pE0`, `pE1` und `pE2` zeigen auf drei aufeinander folgende Elemente. Schreiben Sie die Programmzeilen, die das mittlere der drei Elemente löschen.

```
/* Löschen des Elementes pE1 aus der Kette */
```

3. Aufgabe (8 Punkte)

In dem folgenden Programm fehlen wichtige Teile. Bitte ergänzen.

```
#include <stdio.h>

int main()
{
    FILE *out=NULL;
    int Fehler=0;
    char Dateiname[] = "Textdatei.txt";

    if(!Fehler) fprintf(out, "Heute ist Freitag, der 6.7.2007.\n");

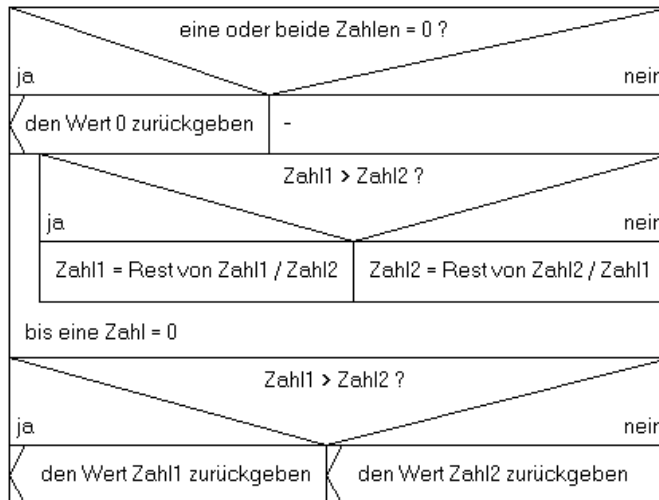
    return Fehler;
}
```

4. Aufgabe Funktion erstellen (18 Punkte)

Eine Funktion wurde wie folgt deklariert:

```
unsigned Funktion(unsigned Zahl1, unsigned Zahl2);
```

Der Inhalt der Funktion ist durch das folgende Struktogramm wiedergegeben:



Erstellen Sie die Definition der Funktion:

```
unsigned Funktion(unsigned Zahl1, unsigned Zahl2)
```

```
{
```

```
}
```

Welche Rückgabewerte ergeben sich?

Zahl1	Zahl2	Rückgabewert
3	6	
8	12	
18	33	

Geben Sie der Funktion einen sinnvollen Namen:

5. Aufgabe Fehlersuche (18 Punkte)

In das Programm auf der folgenden Seite haben sich in 21 Zeilen Fehler eingeschlichen. Tragen Sie die gefundenen Fehler mit dazugehöriger Korrektur in die folgende Tabelle ein. Der erste der 21 Fehler wurde beispielhaft eingetragen.

Zeile	Fehler	Korrektur	
4	<i>Kommentarende fehlt</i>	<i>... 1.00 */</i>	

```

/* 1 */ /*-----*/
/* 2 */ /* Autor      : Dr. Robert Heß */
/* 3 */ /* Dateiname   : EinMalEins.c */
/* 4 */ /* Version     : 1.00
/* 5 */ /* Erstellt    : 16.6.2007 */
/* 6 */ /* Beschreibung: Kleiner Trainer für das große 1x1 */
/* 7 */ /*-----*/
/* 8 */
/* 9 */ #include <stdio.h>
/* 10 */ #include <stdlib.h>
/* 11 */ #include <time.h>
/* 12 */ #include <conio.h>
/* 13 */
/* 14 */ #define ANZAHL 10;      /* Anzahl der Aufgaben pro Runde */
/* 15 */
/* 16 */ int Main()
/* 17 */ {
/* 18 */     int AufgNr;          /* Aktuelle Nummer der Aufgabe */
/* 19 */     int Zahl1; Zahl2;    /* Multiplikatoren */
/* 20 */     int Ergeb;          /* Eingegebenes Ergebnis der Aufgabe */
/* 21 */     int Richtige;       /* Anzahl der richtigen Lösungen */
/* 22 */     char Text[];        /* Eingegebene Textzeile */
/* 23 */     time t StartZeit;  /* Startzeit der aktuellen Runde */
/* 24 */     int Dauer;         /* Benötigte Zeit für Runde */
/* 25 */     int Punkte;       /* erzielte Punktzahl */
/* 26 */     char Taste;       /* betätigte Taste */
/* 27 */
/* 28 */     /* Schleife über mehrere Durchgänge */
/* 29 */     do
/* 30 */     {
/* 31 */         /* Überschrift ausgeben */
/* 32 */         system("cls");
/* 33 */         printf("+-----+\n");
/* 34 */         printf("| Trainieren Sie das gro\341e 1x1! |");
/* 35 */         printf("+-----+\n");
/* 36 */
/* 37 */         /* Startzeit merken und Zufallszahlen initialisieren
/* 38 */         StartZeit == time(NULL);
/* 39 */         srand(StartZeit)
/* 40 */
/* 41 */         /* Lösung der Aufgaben */
/* 42 */         for(AufgNr=1, AufgNr<=ANZAHL, AufgNr++) {
/* 43 */             Zahl1 = rand()%19+2;
/* 44 */             Zahl2 = rand()%19+2;
/* 45 */             printf("%d x %d = ", Zahl1, Zahl2);
/* 46 */             fgets(&Text, 20, stdin);
/* 47 */             if(sscanf(Text, "%d", &Ergeb)==1 && Zahl1*Zahl2==Ergeb)
/* 48 */                 Richtige++;
/* 49 */         }
/* 50 */
/* 51 */         /* Ergebnis bewerten */
/* 52 */         Dauer = time(NULL)-StartZeit;
/* 53 */         printf("Sie haben %d von %d Aufgaben richtig gel\224st.\n",
/* 54 */             Richtige, ANZAHL);
/* 55 */         printf("Das entspricht %lg %%. \n", Richtige/ANZAHL*100.0);
/* 56 */         printf("Sie haben daf\201r %d Sekunden ben\224tigt\n", dauer);
/* 57 */         if(Dauer<0) Dauer = 1;
/* 58 */         Punkte = Richtige*10000/Dauer;
/* 59 */         printf("Damit haben Sie %d Punkte!\n", Punkte);
/* 60 */
/* 61 */         /* neue Runde? */
/* 62 */         printf("\nWollen Sie nochmal (J/N)?\n");
/* 63 */         do {
/* 64 */             Taste = toupper(getch());
/* 65 */             } while(Taste!='J' || Taste!='N');
/* 66 */
/* 67 */         /* Wenn gewünscht, neue Runde */
/* 68 */         } while(Taste=='N');
/* 69 */
/* 70 */         /* Schlussmeldung */
/* 71 */         printf("\nTsch\201\341, bis zum n\204chsten Mal!\n\n");
/* 72 */
/* 73 */         return NULL;
/* 74 */     }

```

6. Aufgabe (8 Punkte)

Eine Klasse soll die dynamische Speicherverwaltung für einen eindimensionalen Vektor von Gleitkommazahlen übernehmen. Die Klasse arbeitet mit den Operatoren `new` und `delete` und wurde wie folgt deklariert:

```
class cData {
private:
    double *data;
    int size;
public:
    cData();
    ~cData();
    bool setSize(int newSize);
    bool setValue(int index, double value);
    double getValue(int index);
};
```

Erstellen Sie die Definition für Konstruktor und Destruktor.