

Klausur PR2

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

Dr. Robert Heß, 1.2.2008

Bearbeitungsdauer: 90 min

Hilfsmittel: Vorlesungsunterlagen und C/C++ Einführungsbücher (z.B. Erlenkötter)
Jegliche Art von elektronischen Hilfsmitteln sind nicht erlaubt.

1. Aufgabe (6 Punkte)

Definieren Sie einen zweidimensionalen Vektor vom Typ `int` mit 8 x 8 Elementen. Der Vektor soll den Namen `Schachbrett` haben.

.....

Definieren Sie einen Zeiger mit dem Namen `pField`, der auf das erste Element der vorher definierten Variable zeigt.

.....

Die Variable `pComplex` ist ein Zeiger auf eine Struktur mit den Elementen `real` und `imag`, die jeweils vom Typ `double` sind. Geben Sie den Realteil (`real`) mit der Funktion `printf()` auf dem Bildschirm aus.

.....

2. Aufgabe (10 Punkte)

Mit welchen Funktionen werden in C Dateien geöffnet und geschlossen?

.....

Mit welchen Operatoren wird in C++ Speicher reserviert und wieder freigegeben?

.....

Auf welche drei Eigenschaften muss bei der rekursiven Programmierung geachtet werden?

1.

2.

3.

3. Aufgabe (12 Punkte)

Reservieren Sie im folgenden Programm für 100 Personen Speicher und weisen Sie der 10. Person das Geburtsdatum *1.3.1985* zu.

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>

typedef struct {
    int Jahr;
    int Monat;
    int Tag;
} tDatum;

typedef struct {
    char Nachname[51];
    char Vorname[51];
    tDatum Geburt;
} tPerson;

#define ANZAHL 100

int main()
{
    tPerson *Person=NULL;

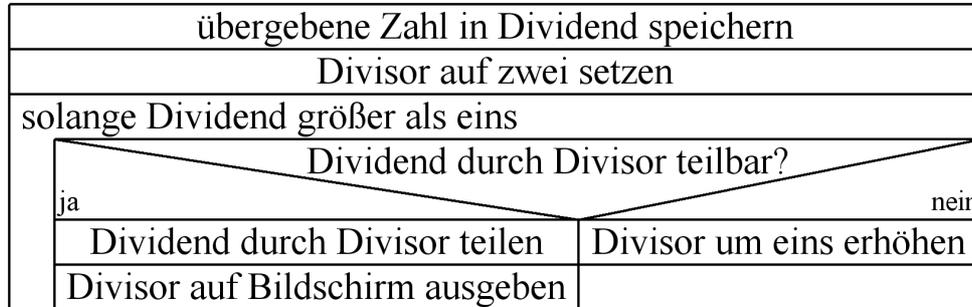
    return 0;
}
```

4. Aufgabe Funktion erstellen (18 Punkte)

Eine Funktion wurde wie folgt deklariert:

```
void Funktion(unsigned Zahl);
```

Der Inhalt der Funktion ist durch das folgende Struktogramm wiedergegeben:



Erstellen Sie die Definition der Funktion:

```
void Funktion(unsigned Zahl)
{
```

}

Welche Ausgabe erzeugt die Funktion?

Zahl	Ausgabe
12	
17	
60	

Geben Sie der Funktion einen sinnvollen Namen:


```
/* 1 */ /*-----*/
/* 2 */ /* Autor : Dr. Robert Heß */
/* 3 */ /* Dateiname : main.c */
/* 4 */ /* Version : 1.00
/* 5 */ /* Erstellt : 9.1.2008 */
/* 6 */ /* Beschreibung: Größter gemeinsamer Teiler von zwei Zahlen */
/* 7 */ /*-----*/
/* 8 */
/* 9 */ #include <stdio.h>
/* 10 */
/* 11 */ int ggt(int Zahl1, int Zahl2);
/* 12 */ int getInt(char *text, int min, int max);
/* 13 */
/* 14 */ int main()
/* 15 */ {
/* 16 */     int Zahl1; /* zwei Zahlen zum Finden */
/* 17 */     int Zahl2; /* des größten gemeinsamen Teilers */
/* 18 */
/* 19 */     /* zwei Zahlen vom Benutzer abfragen */
/* 20 */     Zahl1 = getInt("Erste Zahl", 2, 1000000);
/* 21 */     Zahl2 = getInt("Zweite Zahl", 2, 1000000);
/* 22 */
/* 23 */     /* größten gemeinsamen Teiler ermitteln */
/* 24 */     printf("Größter gemeinsamer Teiler von %d und %d "
/* 25 */           "ist: %d\n", Zahl1, Zahl2, ggt(Zahl1, Zahl2));
/* 26 */
/* 27 */     return NULL;
/* 28 */ }
/* 29 */
/* 30 */ int ggt(int Zahl1; int Zahl2)
/* 31 */ {
/* 32 */     /* nur positive Zahlen bearbeiten */
/* 33 */     if(Zahl1<=0 || Zahl2<=0) return 0;
/* 34 */
/* 35 */     /* größter gemeinsamer Teiler nach dem Prinzip von Euler */
/* 36 */     while(Zahl1 && Zahl2) {
/* 37 */         if(Zahl1>Zahl2) Zahl1 %= Zahl2;
/* 38 */         else Zahl2 %= Zahl1
/* 39 */     }
/* 40 */
/* 41 */     /* größeren Wert zurückgeben
/* 42 */     return Zahl1>Zahl2?Zahl1:Zahl2;
/* 43 */ }
/* 44 */
/* 45 */ /* Liest eine ganze Zahl vom Benutzer ein */
/* 46 */ int getInt(char *text, int min, int max)
/* 47 */ {
/* 48 */     char line[51]; /* Eingabezeile */
/* 49 */     int neuerWert=0; /* neuer Eingabewert */
/* 50 */     int korrekt=0; /* Indikator für korrekte Eingabe */
/* 51 */
/* 52 */     do
/* 53 */         /* Frage an Benutzer ausgeben und Wert abfragen */
/* 54 */         printf("%c (%d - %d): ", text, min, max);
/* 55 */         fgets(line, 50, stdin);
/* 56 */
/* 57 */         /* Test, ob eine Zahl eingegeben wurde */
/* 58 */         if(sscanf(line, "%d", neuerWert)!=1)
/* 59 */             printf("Bitte geben Sie eine Zahl ein!\n");
/* 60 */
/* 61 */         /* Bereich prüfen */
/* 62 */         else if(min>neuerWert>max)
/* 63 */             printf("Zahl außerhalb des gültigen Bereichs!\n");
/* 64 */
/* 65 */         /* die Zahl ist korrekt */
/* 66 */         else korrekt = 1;
/* 67 */
/* 68 */         /* wiederhole bis Zahl korrekt */
/* 69 */     } while(korrekt);
/* 70 */
/* 71 */     /* neuen Wert zurück geben */
/* 72 */     return neuerWert;
/* 73 */ }
```

6. Aufgabe (8 Punkte)

Die Klasse `cKomplex` soll das Arbeiten mit komplexen Zahlen erleichtern. Im Folgenden sehen Sie die Deklaration der Klasse mit drei Methoden, die noch nicht definiert sind.

```
class cKomplex {
public:
    double real;
    double imag;

    cKomplex() {};
    cKomplex(double re, double im);
    void add(double re, double im);
    void sub(double re, double im);
};
```

Implementieren Sie die drei deklarierten Methoden die noch nicht definiert wurden.