

Hochschule Deggendorf Dr. Peter Jüttner	
Vorlesung: Einführung in die Programmierung	WS 2012
Übung 8	Termin 20.12.11

Pointer

1. Wie ist der Inhalt der Variablen a und b nach Ausführung der folgenden C-Code-Sequenzen:

```
int a = 5, b = 10;
int *ptr1, *ptr2;
ptr1 = &a ;
ptr2 = &b;
*ptr1 = *ptr1 + b;
ptr1 = ptr2;
*ptr1 = *ptr1 + b;
```

Überlegen Sie zunächst, schreiben Sie dann ein Programm, das die Werte der Variablen ausgibt.

a hat am Ende den Wert 15, da die Werte von a und b über Pointer ptr1 addiert und an a zugewiesen werden. b hat den Wert 20, da ptr2 auf b zeigt, dann wird ptr2 an ptr1 zugewiesen, d.h. ptr1 zeigt auch auf b, Dann wird b über den ptr2 wie zuvor a verändert dadurch, dass zum dereferenzierten Pointer der Wert von b dazu addiert wird.

2. Schreiben Sie ein Programm, das dynamisch über drei Pointervariable drei Integer auf dem Heap anlegt.
 - a.) Vergleichen Sie die Adressen auf dem Heap, d.h. geben Sie die Inhalte der Pointervariablen aus.
 - b.) Lesen Sie zwei Integerwerte in zwei dieser Heapvariablen ein, addieren Sie die beiden Zahlen, speichern Sie das Ergebnis in der dritten Variablen und geben Sie es aus.
 - c.) geben Sie den Speicher auf dem Heap wieder frei.

```

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

int main()
{
    int *pzahl1, *pzahl2, *pzahl3;

    printf("Addition von zwei Zahlen auf dem Heap\n\n");
    pzahl1 = (int*) malloc(sizeof(int)); /* Speicherplatz auf dem Heap anfordern
    */
    pzahl2 = (int*) malloc(sizeof(int)); /* Speicherplatz auf dem Heap anfordern
    */
    pzahl3 = (int*) malloc(sizeof(int)); /* Speicherplatz auf dem Heap anfordern
    */
    /* Hier müsste eigentlich geprüft werden, ob malloc erfolgreich war */

    /* Ausgabe der Pointer */
    printf("Adresse 1. Pointer %p\n", pzahl1);
    printf("Adresse 2. Pointer %p\n", pzahl2);
    printf("Adresse 3. Pointer %p\n", pzahl3);

    printf("Bitte erste Zahl eingeben\n");
    scanf("%d", pzahl1); /* hier ohne &-Operator ! */
    printf("Bitte zweite Zahl eingeben\n");
    scanf("%d", pzahl2); /* hier ohne &-Operator ! */

    *pzahl3 = *pzahl1 + *pzahl2; /* Addieren der Zahlen auf dem Heap */

    printf("Das Ergebnis von %d + %d ist %d\n\n", *pzahl1, *pzahl2, *pzahl3);

    /* Heap wieder freigeben */
    free(pzahl1); free(pzahl2); free(pzahl3);

    system("PAUSE");
}

```

3. Schreiben Sie ein Programm, das mittels Pointer herausfindet, ob eine Zahl in einem Integer Array der Länge 20 vorkommt. Die 20 Zahlen und die gesuchte Zahl sollen interaktiv eingelesen werden. Arbeiten Sie nicht mit einem Array-Index!

Allgemeines Lösungsprinzip: Ein Vektornamen wird auch als Pointer interpretiert, der auf das erste Element (Index 0) des Vektors zeigt. Interpretiert als Pointer kann über den Vektornamen mittels Pointerarithmetik

und Dereferenzierung (*(vektorname+i)) alternativ zum Zugriff über den Index (vektorname[i]) auf Elemente des Vektors zugegriffen werden.

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

int main()
{
    int zahlenfeld[20];
    int i;
    int zahl;
    int gefunden = 0;

    printf("Einlesen von 20 Zahlen\n");
    for (i = 0; i < 20; i++)
    { printf ("Bitte %d. Zahl eingeben:\n", i+1);
      scanf("%d",zahlenfeld+i); /* Pointerarithmetik */
    };

    printf("Bitte gesuchte Zahl eingeben:\n");
    scanf("%d",&zahl);

    for (i = 0; i < 20; i++)
        if (*(zahlenfeld+i) == zahl) && (gefunden == 0))
            gefunden = i;
    if (gefunden)
        printf ("Zahl gefunden erstmals an Stelle %d\n", gefunden+1);
    else printf ("Zahl nicht gefunden\n");

    system("PAUSE");
}
```