

Übungsblatt 1

Ziel dieses Übungsblattes ist es, dass Sie sich mit den Funktionen und dem Prinzip der dynamischen Speicherallokierung vertraut machen.

Aufgabe 1 – Dynamisches Erzeugen eines Arrays

Gegeben ist folgender lückenhafte Programmausschnitt:

```
int main()
{
    int *ar;
    int i, anzahl;
    printf("Wieviele Zahlen moechten Sie einlesen: ");
    scanf(/*Hier ergänzen*/);

    /*Hier Code zum Erzeugen des dynamischen Int-Arrays*/

    for(i = 0; i < n; i++)
        scanf("%d", /*Hier ergänzen*/);

    /*Hier Code für die Ausgabe ergänzen */

    /*Hier Code zum Freigeben des Speichers ergänzen */

    return 0;
}
```

- Schauen Sie sich das obige Programm an und versuchen Sie den Ablauf zu verstehen.
- Ergänzen Sie die fehlenden Stellen im Programm, so dass die eingelesenen n Zahlen in einem dynamischen Feld gespeichert werden und anschließend untereinander ausgegeben werden.

Aufgabe 2 – Dynamisches Erzeugen von Strukturen

Gegeben ist folgender lückenhafte Programmausschnitt:

```
typedef struct {
    float x;
    float y;
} coordinate_t;

coordinate_t* getCoordinateArray(int n) {
    /* Hier ergänzen */
}

int main(void)
{
    coordinate_t *q;
    int i = 0;

    q = getCoordinateArray(10);

    for(i=0; i<10; i++)
        printf("%f,%f\n", q[i].x, q[i].y);
    free(q);
    return 0;
}
```

Die Funktion `getCoordinateArray` soll auf dem Heap ein Feld zur Speicherung von Koordinaten (Datentyp `coordinate_t`) erzeugen und einen Zeiger auf das erste Feldelement zurückliefern. Die Feldgröße `n` wird beim Aufruf übergeben. Die x- und y-Koordinaten der Feldelemente sollen mit 0 initialisiert werden. Ergänzen Sie die fehlenden Zeilen und testen Sie ihr Programm aus!

Aufgabe 3 – Zeichenketten verknüpfen

Gegeben sei folgender Programmausschnitt:

```
char* concatenate(char*, char*);

int main()
{
    char str1[] = "Umwelt-";
    char str2[] = "Campus";

    char *erg = concatenate(str1, str2);
    printf("Ergebnis: %s", erg);

    free(erg);
    return 0;
}
```

Die Funktion *concatenate()* erwartet als Argument zwei Zeiger auf eine Zeichenkette. Die Funktion soll einen Zeiger auf eine Zeichenkette zurückliefern, die aus Aneinanderhängen der ursprünglichen Zeichenketten entsteht. Die Funktion soll hierzu ein char-Feld auf dem Heap, in dem die neue Zeichenkette gespeichert wird. Die Adresse des Feldes wird als Ergebnis zurückgeliefert. Ergänzen Sie die fehlenden Zeilen!

Hinweis: Die Funktion *strlen* liefert die Länge eines Strings (ohne Null-Byte) zurück.

Aufgabe 4 – Fehlersuche

Schauen Sie sich folgendes Programm an. Welche Zeilen enthalten Fehler?

```
int main()
{
    int *zeiger;
    int *zeiger2;
    int *zeiger3;
    int array[10];

    zeiger = (int *) malloc(sizeof(*zeiger) * 10);
    zeiger2 = zeiger;
    zeiger3 = zeiger++;
    free(zeiger);
    free(zeiger3);
    free(zeiger2);
    free(array);
}
```