Lösungen zur Vorlesung
GRUNDLAGEN DER INFORMATIK I
Studiengang Elektrotechnik
WS 02/03

AG Betriebssysteme FB3
Kirsten Berkenkötter

Lösung Übungszettel 2

1 Aufgabe 1

```
//Zuerst muss die Header-Datei stdio.h eingebunden werden
//Diese stellt z.B. printf zur Verfügung
#include <stdio.h>

//es folgt das Hauptprogramm
int main()
{
   //zuerst werden zwei Variablen deklariert
   //die Variable i wird im Programm als Zähler für den Index (for-Schleife) verwendet
   //die Variable max wird als Merker für den Index verwendet, an dem das Maximum liegt
   int i, max;
   //ein Array vom Typ integer wird als Feld verwendet, in dem gesucht werden soll
   //das Feld wird mit festen integer-Werten vorbelegt
   int zahlenfeld[] = {-4, 155, -32769, 8, 32768, 3333, -4949, 4, 111, -99};

   //zunächst wird das Maximum auf 0 gesetzt, um einen Vergleichswert zu haben
   max = 0;

   //in der for-Schleife wird nach dem Maximum gesucht
   for (i = 0; i < 10; i++)
   {
      //in diesr Bedingung wird abgefragt, ob das aktuelle Feld des Arrays einen größeren
      //Wert enthält als das bisherige Maximum
      //wenn dies der Fall ist, wird der aktuelle Index als neuer Maximalwert gespeichert
      //Achtung: Bei mehrfachem Auftreten des Maximalwerts im Array wird immer nur der
      //Index des ersten Auftretens gespeichert
      if (zahlenfeld[i] > zahlenfeld[max])
```

```
        max = i;
    }

    //Der Maximalwert wird ausgegeben
    printf("Maximalwert: %d", zahlenfeld[max]);
    //Der Index des (ersten Auftretens) des Maximalwerts wird ausgegeben
    printf(" an Stelle %d\n", max);
}
```

2 Aufgabe 2

```
#include <stdio.h>

int main()
{
    //Variable für den Zähler in der Schleife
    int i;
    //Variable für die Zahl, nach der gesucht werden soll
    int zahl;
    //Array für die Zahlen, in denen gesucht werden soll
    int zahlenfeld[] = {-4, 155, -3199, 8, 31999, 3333, -4949, 4, 111, -99};

    //eine Zahl festsetzen, nach der gesucht werden soll
    zahl = -99;

    //Zunächst die While-Schleife:
    //als erstes den Zähler auf 0 setzen (Initialisierung)
    i = 0;

    //das Array wird durchsucht, bis entweder jede Zahl geprüft wurde (i < 10)
    //oder die Zahl gefunden wurde (zahlenfeld[i] != zahl
    while ((i < 10) && (zahlenfeld[i] != zahl))
    {
        //der Zähler wird erhöht, um jede Zahl im Feld zu prüfen
        //und um keine Endlosschleife zu programmieren!)
        i++;
```

```
  }

  //wenn i den Wert 10 hat, dann wurde das Feld erfolglos durchsucht (Schleife ist bis
  //zum Ende gelaufen)
  //in diesem Fall, gib die Fehlermeldung aus
  //bei jedem anderen Wert von i wurde die Zahl gefunden und der Index im Feld kann
  // ausgegeben werden
  if (i == 10)
    printf("While-Schleife: Nicht gefunden!\n");
  else
    printf("While-Schleife: Index %d\n", i);

  //Jetzt die For-Schleife
  //durchsuche jedes Feld mit der For-Schleife (Index von 0 bis 9)
  for (i = 0; i < 10; i++)
  {
    //wenn die Zahl gefunden wird, gib den Index aus und unterbreche die Schleife mit break
    if (zahlenfeld[i] == zahl)
    {
      printf("For-Schleife: Index %d\n", i);
      break;
    }
    //diese Stelle wird nur erreicht, wenn die Zahl auch im letzten Feld nicht gefunden wurde
    //(durch das break im Erfolgsfall ist das nicht möglich!)
    //gib also die Fehlermeldung aus
    if (i == 9)
      printf("For-Schleife: Nicht gefunden!\n");
  }
}
```

3 Aufgabe 3

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

//definiere Größe des Arrays
#define MAXNUM 10000



//Funktion für die binäre Suche
//suchzahl ist die gesuchte Zahl, *array ein Zeiger auf das zu durchsuchende Array
//laenge gibt die Länge dieses Arrays an, *zaehler ist ein Zeiger auf einen Zähler,
//der die Anzahl der benötigten Vergleiche zählt
int binaereSuche(int suchzahl,  int *array, int laenge, long *zaehler)
{
   //Variable für die obere Grenze, die untere Grenze und die Mitte
   int obereGrenze, untereGrenze, mitte;

   //obere Grenze am Anfang auf dem obersten Index des Arrays
   //untere Grenze zu Anfang auf 0, unterste Grenze des Arrays
   obereGrenze = laenge-1;
   untereGrenze = 0;

   //nun die binäre Suche
   //solange die obere Grenze größer oder gleich der unteren Grenze ist,
   //ist die binäre Suche gültig
   //ist die obere Grenze kleiner als die untere, wurde das Array erfolglos
   //durchsucht
   while(obereGrenze >= untereGrenze)
   {
      //stelle die Mitte des Teilbereichs fest
      //erhöhe den Zähler für die Anzahl der Vergleiche
      mitte = (obereGrenze + untereGrenze)/2;
      (*zaehler)++;

      //wenn die gesuchte Zahl kleiner als die Zahl in der Arraymitte ist,
```

```
        //dann liegt sie in der unteren Hälfte des Teilbereichs
        //der neue Suchbereich ist dann die untere Hälfte, d.h. neue obere Grenze ist mitte-1
        if (suchzahl < array[mitte])
            obereGrenze = mitte-1;
        //wenn die gesuchte Zahl größer als die Zahl in der Arraymitte ist,
        //dann liegt sie in der oberen Hälfte des Teilbereichs
        //der neue Suchbereich ist dann die obere Hälfte, d.h. neue untere Grenze ist mitte+1
        else if (suchzahl > array[mitte])
            untereGrenze = mitte+1;
        //wenn die gesuchte Zahl gleich der Zahl in der Arraymitte ist,
        //habe ich sie gefunden und gebe den Index im Array zurück
        else
            return mitte;
    }

    //wenn die Schleife durchlaufen ist und die Funktion immer noch läuft (keine Rückgabe),
    //dann ist die Zahl nicht im Array enthalten, Rückgabewert in diesem Fall ist -1
    return -1;
}

int main()
{
    //eine Variable für das zu durchsuchende Array
    //eine Variable für die zu suchende Zahl
    //eine Variable für das Ergebnis (Index der zu suchenden Zahl)
    //eine Variable als Index für die for-Schleife
    //eine Variable als Zähler für die benötigten Vergleiche beim Suchen
    int array[MAXNUM];
    int suchzahl, ergebnis, i;
    long zaehler = 0;

    //belege das int-Array in einer for-Schleife
    //damit ist es auch automatisch sortiert
    for (i = 0; i < MAXNUM; i++)
        array[i] = i;
```

```
    //bestimme eine Zufallszahl als zu suchende Zahl
    srand(0);
    suchzahl = rand() % MAXNUM;

    //suche die Zahl, Ergebnis ist Index dieser Zahl im Array, wenn sie vorhanden ist
    //wenn sie nicht vorhanden ist, ist das Ergebnis -1
    ergebnis = binaereSuche(suchzahl, array, MAXNUM, &zaehler);

    //gib das Ergebnis und die Anzahl der benötigen Vergleiche oder eine Fehlermeldung aus
    if (ergebnis == -1)
        printf("Nicht gefunden\n");

    else
    {
        printf("Zahl %d ist an Stelle %d\n", suchzahl, ergebnis);
        printf("Vergleiche %d\n", zaehler);
    }
}
```