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

AG Betriebssysteme FB3
Kirsten Berkenkötter

Lösung Praktikum 6

1 Aufgabe 1: Zufallszahlen, Sortieren und Suchen

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

//Vergleichsfunktion für qsort and bsearch
//wenn zahl1 < zahl2 => -1
//wenn zahl1 == zahl2 => 0
//wenn zahl1 > zahl2 => 1
int vergleich(const void *zahl1, const void *zahl2)
{
   if (*(int *)zahl1 < *(int *) zahl2)
      return -1;
   else if (*(int *)zahl1 == *(int *) zahl2)
      return 0;
   else
      return 1;
}

int main()
{
   //Array für die Zahlen
   int array[10];
   //Zähler
   int i;
   //zu suchende Zahl
   int suchzahl;
   //Zeiger auf Suchergebnis
   int *ergebnis;

   //Zufallszahlengenerator initialisieren
   srand(time(NULL));

   //Array mit Zufallszahlen füllen
   for(i = 0; i < 10; i++)
      array[i] = rand()%100+1;

   //Array ausgeben
   printf("Array vorher: \n");
   for(i = 0; i < 10; i++)
```

```
    printf("%d. %d\n", i+1, array[i]);

  //Array sortieren
  qsort(array, 10, sizeof(int), &vergleich);

  //Array ausgeben
  printf("Array nachher: \n");
  for(i = 0; i < 10; i++)
    printf("%d. %d\n", i+1, array[i]);

  //zu suchende Zahl bestimmen
  suchzahl=rand()%100+1;

  //Zahl suchen
  ergebnis = (int *)bsearch(&suchzahl, array, 10, sizeof(int), &vergleich);

  //Zahl gefunden => ausgeben
  if(ergebnis != NULL)
    printf("%d gefunden %d\n", suchzahl, *ergebnis);
  //Zahl nicht gefunden => ausgeben
  else
    printf("%d nicht gefunden!\n", suchzahl);
}
```

2 Aufgabe 2: Zeichenketten, Sortieren und Suchen

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

//Vergleichsfunktion für Quicksort und binäre Suche
//vergleicht Zeichenketten
int vergleich(const void *kette1, const void *kette2)
{
  //wenn der erste Buchstabe der ersten Kette
  //kleiner ist als der erste der zweiten
  //-1 (kette1 kleiner als kette2)
  if(*(char *)kette1 < *(char *)kette2)
  {
    return -1;
  }
  //wenn der erste Buchstabe der ersten und zweiten
  //Kette gleich sind,
  //müssen die weiteren Buchstaben überprüft werden
  else if(*(char *)kette1 == *(char *)kette2)
  {
    //solange etwas in der Zeichenkette steht, überprüfen
    while(*(char *)kette1)
    {
      //nächstes Zeichen vergleichen
      kette1 ++;
```

```
            kette2 ++;
            //wenn das nächste Zeichen der ersten Kette
            //kleiner als nächste der zweiten Kette
            //-1 (kette1 kleiner als kette2
            if(*(char *)kette1 < *(char *)kette2)
               return -1;
            //immer noch gleich, weiter überprüfen
            else if(*(char *)kette1 == *(char *)kette2)
            {
               ;
            }
            //wenn das nächste Zeichen der ersten Kette
            //größer als das nächste der zweiten
            //1 (kette1 größer als kette2
            else
               return 1;
         }
         //alle Buchstaben überprüft und gleich
         //0 (kette1 == kette2)
         return 0;
      }
      //wenn der erste Buchstabe größer als der zweite ist
      //1 (kette1 größer als kette2)
      else
         return 1;
}

int main()
{
      //ein Array mit 10 Einträgen
      //die Einträge sind Zeichenketten der Länge 20
      char array[10][20];
      //Zähler
      int i;
      //ein Name zum Suchen
      char name[20];
      //ein Zeiger für das Ergebnis der Suche
      char *ergebnis;

      //zehn Namen einlesen
      for(i = 0; i<10; i++)
      {
         printf("Gib einen Namen ein: ");
         scanf(" %s", &(array[i]));
      }

      //Liste zum ersten Mal ausgeben
      printf("\nListe vorher: \n");
      for(i = 0; i < 10; i++)
         printf("%i. %s\n", i+1, array[i]);
```

```
    //Liste sortieren
    qsort(array, 10, sizeof(char)*20, &vergleich);

    //Liste zum zweiten Mal ausgeben
    printf("\nListe nachher: \n");
    for(i = 0; i < 10; i++)
        printf("%d. %s\n", i+1, array[i]);

    //Welche Person soll gesucht werden?
    printf("\nWen suchen: \n");
    scanf("%s", &name);

    //Person suchen
    ergebnis = (char *)bsearch(name, array, 10, sizeof(char)*20, &vergleich);

    //Person gefunden
    if(ergebnis != NULL)
        printf("%s gefunden!\n", ergebnis);
    //Person nicht gefunden
    else
        printf("Nicht gefunden!\n");
}
```