

# Angaben zur 6. Übung am 30. 11. 2010

## Beispiel

Schreiben Sie ein C-Programm das:

1. Die Werte für Ordnungszahl, Name, Atomgewicht aus einer Datei in ein Array einliest und den Speicherplatz dynamisch reserviert.
2. Zwei Zahlenwerte von der Kommandozeile übernimmt und alle Daten der Elemente ausgibt, deren Atomgewicht zwischen diesen beiden Werten liegt.
3. Einen String von der Kommandozeile übernimmt und alle Daten für das übereinstimmende Element ausgibt.
4. Wird kein Parameter angegeben, soll die Tabelle nach dem Namen sortiert ausgegeben werden. (Nicht Pflicht)

Zerteilen sie das Programm in Unterprogramme (Funktionen)

1. `struct Elem * Read(char * FileName, int * length);`  
Belegen des Speicherplatzes, Lesen der Werte in ein Array, Rückgabe des Zeigers
2. `int SearchWeight(struct Elem *p, int length, double w1, double w2);`  
Suchen und Ausgeben der Elemente zwischen w1 und w2
3. `struct Elem SearchElem(struct Elem *p, int length, char * e);`  
Suchen des passenden Elements e
4. `void Sort(struct Elem *p, int length);`  
Sortieren

Das Hauptprogramm kann etwa so aussehen:

```
struct Elem{    int Z;  
               char Name[3];  
               double W;  
};
```

```
int main(int argc, char **argv)
```

```

{
  struct Elem * table;
  char *ElemFile="elem.dat";

  int nElem;

  /* Pruefen der Anzahl der Kommandozeilenparameter */
  if (argc > 3)
    {printf("Usage: %s [w1 w2] or [name] or [none]\n",argv[0]);
      exit(1);
    }

  /* Einlesen der Tabelle */
  table=Read(char * ElemFile,&nElem);
  if(!table)
    {printf("Illegal table\n");
      exit(1);
    }

  if (argc==3)
    {/* Suchen:
       Kommandozeilenparameter sind 2 Zahlen */
      double w1=atof(argv[1]);
      double w2=atof(argv[2]);
      int n;
      n=SearchWeight(table,nElem,w1,w2);
      printf("%d records found\n",n);
    }

  if (argc==2)
    {/* Suchen:
       Kommandozeilenparameter ist ein String */
      struct Elem result;
      result=SearchElem(table,nElem,argv[1]);
      printf("Found:\n");
      printf("%d %s %f\n",result.Z,result.Name,result.W);
    }

  if (argc==1)
    {/* Sortierte Ausgabe:*/
      int i;
      Sort(table,nElem);
    }
}

```

```
        for(i=0;i<nElem;i++)
            printf("...",...);
    }

    /* Freigabe des zuvor in Read(..) reservierten Speichers */
    free(table);

    exit(0);
}
```

## Anleitung

Während der Übung ist ein kurzes Protokoll anzufertigen und als `PROTOKOLL.txt` im Verzeichnis des jeweiligen Übungstages abzuspeichern.

z.B. `~/06Ue2010-11-30/PROTOKOLL.txt`

Das Protokoll ist eine einfache ASCII-Text-Datei, die mit einem Text-Editor mit dem Sie auch Ihre Programme schreiben, erstellt wird. Nennen Sie diese Datei unbedingt

`PROTOKOLL.txt`.

Das Protokoll muss folgendes enthalten:

1. Datum, Übungsnummer, Gruppennummer, Name(n) der mitwirkenden StudentInnen
2. Benötigter Zeitaufwand für die gestellten Aufgaben (circa)
3. Namen der erstellten Programme (KEINE Listings). Die erstellten Dateien müssen sich ebenfalls im oben genannten Verzeichnis befinden.
4. Kurze Antwort auf eventuell oben gestellte Fragen
5. Eventuelle Probleme oder Besonderheiten, falls diese aufgetreten sind.