```c
/* Aufgabe 1 Blatt 7 */
#include <stdio.h>
#include <stdlib.h>



int compar(const void *, const void *);


int main()
{
  int anzahl;
  int zaehl;
  float *zahlen;


  printf("Anzahl eingeben: ");
  scanf("%d", &anzahl);

  /* Platz fuer die Zahlen schaffen */
  zahlen = (float *)malloc(anzahl*sizeof(float));
  if (zahlen == NULL)
  {
    printf("Fehler\n");
    exit(-1);
  }

  /* Zahlen einlesen */
  for (zaehl=0; zaehl<anzahl; zaehl++)
  {
    printf("Bitte %d. Zahl eingeben: ", zaehl);
    scanf("%f", &zahlen[zaehl]);
  }

  printf("Feld ausgeben\n");
  for (zaehl=0; zaehl<anzahl; zaehl++)
    printf("%f\n", zahlen[zaehl]);

  /* Und hier wird sortiert */
  qsort(zahlen, anzahl, sizeof(float), compar);

  printf("Feld ausgeben\n");
  for (zaehl=0; zaehl<anzahl; zaehl++)
    printf("%f\n", zahlen[zaehl]);

  return 0;
}
```



```c
int compar(const void *a, const void *b)
{
  if (*(float *)a < *(float *)b)  return 1;
  if (*(float *)a > *(float *)b)  return -1;
  return 0;
}


/* ---------------------------------------------------------- */
/* Musterloesung Uebung 7, Aufgabe 2 */
/* ---------------------------------------------------------- */

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


void StackTiefenTest(int top)
{
  if (top < 2)
  {
    printf("Fehler: Nicht genuegend Argumente auf dem Stack\n");
    exit(-1);
  }
}


int main ( int argc , char ** argv )
{
  double * stack ;
  int i, top = 0 ;

  /* Stack fuer die Eingabe und Zwischenergebnisse */
  /* der Stack kann hoechstens so gross wie die Eingabe werden */
  stack = (double *) malloc ( argc * sizeof ( double ) ) ;

  /* jetzt wird die Eingabe von links nach rechts abgearbeitet */
  for ( i = 1 ; i < argc ; i++ )
  {
    /* Teste das erste Zeichen des momentanen Arguments */
    /* Wenn es ein Operator (+-x/) ist, dann verwende die beiden */
    /* obersten Stackelemente als Operanden */
    /* Ansonsten haben wir eine Zahl und diese wird auf den Stack gelegt */
    switch ( argv[i][0] )
    {
    case '+' :
      StackTiefenTest(top);
      stack[top-2] += stack[top-1] ;
      top -- ;
      break  ;
```

```
    case '-' :
      StackTiefenTest(top);
      stack[top-2] -= stack[top-1] ;
      top -- ;
      break ;
    case 'x' :
      StackTiefenTest(top);
      stack[top-2] *= stack[top-1] ;
      top -- ;
      break ;
    case '/' :
      StackTiefenTest(top);
      stack[top-2] /= stack[top-1] ;
      top -- ;
      break ;
    default :
      /* Zahl gefunden: Umwandeln und auf den Stack legen */
      stack[top] = atof ( argv[i] ) ;
      top ++ ;
      break ;
    }
  }

  if (top > 1)
    printf("Fehler: Noch verbleibende Argumente auf dem Stack\n");

  printf ( "Ergebnis : %.3f\n" , stack[0]) ;

  free ( stack ) ;

  return 0 ;
}
```