```c
/************************************************************
 * Uebung 8, Aufgabe 1
 * minmax
 ************************************************************/

#include <stdio.h>

#ifdef MAX
int max(int x, int y)
{
  /* Zugegeben etwas kryptisch, aber kurz, praegnant und elegant */
  return (x>y) ? x : y;
}

#else
int min(int x, int y)
{
  return (x<y) ? x : y;
}
#endif

main()
{
  int a, b;

  printf("Bitte 2 Zahlen eingeben: ");
  scanf("%d %d",&a, &b);
#ifdef MAX
  printf("Ergebnis: %d\n", max(a, b));
#else
  printf("Ergebnis: %d\n", min(a, b));
#endif

  return 0;
}
/************************************************************
 * Uebung 8, Aufgabe 2
 * str_cat.c
 ************************************************************/

void str_cat(char *string1, char *string2)
{
  /* zuerst das Ende des ersten Strings suchen */
  while (*string1 != '\0') string1++;

  /* an dieser Stelle den zweiten String anhaengen */
  while (*string2 != '\0')
  {
    *string1 = *string2;
    string1++;
```



```c
    string2++;
  }

  *string1 = '\0';              /* Ende anhaengen */
}
/************************************************************
 * Uebung 8, Aufgabe 2
 * str_cat.h
 ************************************************************/

#ifndef _STR_CAT_H_
#define _STR_CAT_H_

void str_cat(char *, char *);

#endif
/************************************************************
 * Uebung 8, Aufgabe 2
 * test.c
 ************************************************************/
#include<stdio.h>
#include "str_cat.h"

#define MAXSTRLEN 50

int main(void)
{
  char str1[MAXSTRLEN];
  char str2[MAXSTRLEN];
  char *strpntr1;
  char *strpntr2;

  /* String einlesen */
  printf("Bitte geben sie einen String (max. 25 Zeichen) ein: ");
  scanf("%s", str1);
  printf("Bitte geben sie noch einen String (max. 25 Zeichen) ein: ");
  scanf("%s", str2);

  /* Zeiger setzen */
  strpntr1 = &str1[0];
  strpntr2 = &str2[0];

  /* Strings verbinden */
  str_cat(strpntr1, strpntr2);

  printf("Der zusammengesetzte String lautet %s.\n", str1);

  return 0;
}
#include <stdio.h>
```

```
#include "ladefile.h"


int LoadFile(char *name, char *buffer, int bufsize)                LoadFile
{
  FILE *filepntr;
  char *bufferpntr;
  int filesize=0;


  bufferpntr = &buffer[0];

  filepntr = fopen(name, "r");
  if (filepntr == NULL)
  {
    printf("Konnte Datei %s nicht oefnnen.\n", name);
    exit(-1);
  }

  while ((!feof(filepntr)) && (filesize < bufsize))
  {
    *bufferpntr = fgetc(filepntr);
    bufferpntr++;
    filesize++;
  }

  if (filesize >= bufsize)
  {
    printf("Datei zu gross.\n");
    exit(-1);
  }

  fclose(filepntr);

  return filesize-1;
}
/************************************************
 * Uebung 8, Aufgabe 3
 *
 ************************************************/

#include <stdio.h>

#define MAXBUFFER 100000

/* Funktion zum Einlesen der Daten */

int LoadFile(char *name, char *buffer, int bufsize)                LoadFile
{
```



```
  FILE *filepntr;
  char *bufferpntr;
  int filesize=0;


  bufferpntr = &buffer[0];

  filepntr = fopen(name, "r");
  if (filepntr == NULL)                   /* falls Oeffnen mislungen */
  {
    printf("Konnte Datei %s nicht oefnnen.\n", name);
    exit(-1);
  }


  /* lese Daten bis Dateiende nicht erreicht
     und Lesepuffer nicht voll */

  while ((!feof(filepntr)) && (filesize < bufsize))
  {
    *bufferpntr = fgetc(filepntr);
    bufferpntr++;
    filesize++;
  }

  /* Programm beenden, falls Puffer zu klein */

  if (filesize >= bufsize)
  {
    printf("Datei zu gross.\n");
    exit(-1);
  }

  fclose(filepntr);    /* Datei schliessen */

  return filesize-1;
}


/* Gib die beiden gelesenen Puffer alternierende zeichenweise aus */

void Vereine(char *buffer1, int filesize1, char *buffer2, int filesize2)    Vereine
{
  int i=0;
  int filesize = filesize1 + filesize2;   /* Gesamtanzahl gelesener Zeichen */

  /************************************************************
   * Die folgende Schleife gibt alle Buchstaben aus.
   * Dabei werden bei geradzahligen Schleifendurchlaeufen jeweils
   * ein Buchstaben vom ersten Puffer, bei ungeradzahligen
```

```
 * Buchstaben vom zweiten Puffer ausgegeben.
 ***************************************************************/

  while (i<filesize)
  {
    if ((i%2)==0)                     /* gerade */
    {
      printf("%c", *buffer1);
      buffer1++;
    }
    else                              /* ungerade */
    {
      printf("%c", *buffer2);
      buffer2++;
    }
    i++;
  }

  printf("\nDas wars...\n");          /* Das wars... :) */
}



/* Das Hauptprogramm */


int main(int argc, char *argv[])
{
  int i=0;
  int filesize1, filesize2;
  char buffer1[MAXBUFFER];
  char buffer2[MAXBUFFER];

  /*******************************************************
   * Das Hauptprogramm erwartet 3 Argumente. Das erste
   * Argument ist der Programmname, die letzten
   * beiden sind die Namen der zulesenden Dateien.
   *******************************************************/

  if (argc != 3)
  {
    printf("Falsche Anzahl von Argumenten.\n");
    exit(-1);
  }

  /*******************************************************
   * lese beide Dateien in die buffer1, buffer2
   * und speichere die Anzahl gelesener Zeichen
   * in filesize1 und filesize2
   *******************************************************/
```

main


```
  filesize1 = LoadFile(argv[1], buffer1, MAXBUFFER);
  filesize2 = LoadFile(argv[2], buffer2, MAXBUFFER);
  printf("Zwei Dateien im Gesamtwert von %d bytes + %d bytes eingelesen.\n", filesize1, filesize

  /* der Unterschied der Dateilaengen darf nicht groesser sein als 1 */

  if (abs(filesize1 - filesize2) > 1)
  {
    printf("Error: verschiedene Dateigroessen.\n");
    exit(-1);
  }

  /* Die spezielle Ausgaberoutine aufrufen */

  Vereine(buffer1, filesize1, buffer2, filesize2);

  return 0;
}
```