```c
/************************************************************
 * Uebung 5, Aufgabe 1                                      *
 * Addition zweier Zahlen mittels Zeigern                   *
 ************************************************************/
#include<stdio.h>


int main(void)
{
  float fliess1, fliess2;
  float *fliesspntr1, *fliesspntr2;
  float summe;

  /* Zahlen einlesen */
  printf("Bitte geben sie eine float Zahl ein: ");
  scanf("%f", &fliess1);
  printf("Bitte geben sie noch eine float Zahl ein: ");
  scanf("%f", &fliess2);

  /* Zeiger auf diese Zahlen erzeugen */
  fliesspntr1 = &fliess1;
  fliesspntr2 = &fliess2;

  /* mittels der Zeiger die zwei Zahlen addieren */
  summe = *fliesspntr1 + *fliesspntr2;

  printf("Die Summe der beiden Zahlen ist ungefaehr %5.2f.\n", summe);

  return 0;
}


/************************************************************
 * Uebung 5, Aufgabe 2                                      *
 * Ausgabe von Zeigerinformationen                          *
 ************************************************************/
#include<stdio.h>

/* -----------------------------
  Funktionen
  ----------------------------- */
void func(unsigned int integer, unsigned int *integerpntr)
{
  printf("Wert des Integers                         \t %u\n", integer);
  printf("Speicherstelle des Integers               \t %u\n", &integer);
  printf("Wert des Integerzeigers                   \t %u\n", integerpntr);
  printf("Wert der Speicherstelle auf die der Integerzeiger zeigt \t %u\n", *integerpntr);
  printf("\n");
}

```



```c


int main(void)
{
  unsigned int integer;
  unsigned int *integerpntr;

  /* Zahlen einlesen */
  printf("Bitte geben sie eine integer Zahl ein: ");
  scanf("%d", &integer);

  /* Zeiger auf die Zahl erzeugen */
  integerpntr = &integer;

  printf("Werte im Hauptprogramm:\n");
  printf("Wert des Integers                         \t %u\n", integer);
  printf("Speicherstelle des Integers               \t %u\n", &integer);
  printf("Wert des Integerzeigers                   \t %u\n", integerpntr);
  printf("Wert der Speicherstelle auf die der Integerzeiger zeigt \t %u\n", *integerpntr);
  printf("\n");

  /* func aufrufen */
  printf("Werte in der Funktion:\n");
  func(integer, integerpntr);

  return 0;
}


/************************************************************
 * Uebung 5, Aufgabe 3                                      *
 * str_cat                                                  *
 ************************************************************/
#include<stdio.h>

#define MAXSTRLEN 50

/* -----------------------------
  Funktionen
  ----------------------------- */
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 */
}



int main(void)                                                         main
{
  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;
}


/************************************************************
 * Uebung 5, Aufgabe 4                                      *
 * Woerter zaehlen                                          *
 ************************************************************/
#include<stdio.h>

#define MAXBUFFERSIZE 1000

/* ------------------------------------------------------------
  Funktionen
  ------------------------------------------------------------ */
/* ************************************************************
  IstTrennzeichen
  liefert eine 1 zurueck, wenn das uebergebene Zeichen ein
  Return, ein Tab oder ein Leerzeichen ist
************************************************************ */
```



```c
int IstTrennzeichen(char zeichen)                                      IstTrennzeichen
{
  if ((zeichen == '\n') || (zeichen == '\t') || (zeichen == ' ')
   || (zeichen == '.')  || (zeichen == ',')  || (zeichen == '!')
   || (zeichen == ';'))
    return 1;

  return 0;
}


/* ************************************************************
  ZaehleWorte
  zaehlt die Worte im buffer der Groesse size
  Idee: Ein Wort endet immer mit einem Buchstaben und einem
  Trennzeichen. Diese Kombinationen werden gezaehlt.
  Also merke ich mir den vorherigen Buchstaben und den
  momentanen. Wenn der vorherige Buchstabe kein Trennzeichen
  war und der momentane Buchstabe ein Trennzeichen, dann
  habe ich das Ende eines Wortes erreicht.
************************************************************ */
int ZaehleWorte(char *buffer, int size)                                ZaehleWorte
{
  int i;
  int worte=0;                  /* Anzahl der Worte */
  char vorherigesZeichen;       /* das Zeichen vor dem momentan bearbeiteten */

  /* das erste Zeichen wird ueberlesen, da vorher noch keine Zeichen gab */
  vorherigesZeichen=*buffer;
  buffer++;

  /* solange noch Buchstaben da sind */
  for (i=1; i<size; i++)
  {
    if (!IstTrennzeichen(vorherigesZeichen) && IstTrennzeichen(*buffer)) worte++;
    vorherigesZeichen = *buffer;
    buffer++;
  }

  return worte;
}



int main(void)                                                         main
{
  char buffer[MAXBUFFERSIZE];  /* Puffer fuer die gelesene Datei */
  int filesize=0;              /* Groesse der Datei */
  FILE *filepntr;              /* Zeiger auf die FILE Struktur */
```

```c
  /* Datei oeffnen und testen, ob es geklappt hat */
  filepntr = fopen("testdatei", "r");
  if (filepntr == NULL)
  {
    printf("Error: Kann Datei nicht oeffnen\n");
    exit(-1);
  }

  /* Datei einlesen */
  for (filesize=0; filesize<MAXBUFFERSIZE; filesize++)
  {
    buffer[filesize] = (char)fgetc(filepntr);  /* einen char einlesen */
    if (feof(filepntr)) break;                  /* Ende der Datei */
  }

  printf("Habe Datei mit Groesse %d Bytes eingelesen.\n", filesize);
  printf("Anzahl der Worte: %d\n", ZaehleWorte(buffer, filesize));

  return 0;
}
```