

Aufgabe 1 (i2s001)

1. Schreiben Sie eine C-Funktion `cmp(op1,op2)`, die zwei ganzzahlige, positive Dezimalzahlen beliebiger Länge miteinander vergleicht. Die beiden Zahlen sollen der Funktion in zwei Textargumenten (Strings aus Ziffern) übergeben werden. Die Funktion soll einen ganzzahligen Wert `w` als Returnwert zurückgeben. Sie soll $w = 0$ zurückgeben, wenn die Zahlen gleich sind und einen Wert $w < 0$ bzw. $w > 0$, wenn $op1 < op2$ bzw. $op1 > op2$ gilt.

Sie können davon ausgehen, daß die Dezimalzahlen keine führenden Nullen enthalten.

2. Schreiben Sie ferner ein Hauptprogramm, das zwei ganzzahlige, positive Dezimalzahlen mit bis zu 80 Ziffern einliest und mit Hilfe Ihrer Funktion `cmp` vergleicht. Zuvor sind die beiden Eingaben zu prüfen, ob es sich um ganzzahlige, positive Dezimalzahlen ohne führende Nullen handelt. Gegebenenfalls ist das Programm mit einer Fehlermeldung abzubrechen. Wurden die Zahlen korrekt eingegeben, so ist das Ergebnis (Returnwert) des Vergleichs auszugeben.

Aufgabe 2 (i2s002)

1. Definieren Sie einen Datentyp `STUDENT`, der geeignet ist, den Namen (Familiennamen) und die Matrikelnummer (`unsigned long`) aufzunehmen. Die Länge des Names sei im Datentyp noch nicht festgelegt.
2. Schreiben Sie eine Funktion `lies(n)`, die ein Feld mit `n` Komponenten vom Typ `STUDENT` erzeugt und einen Zeiger darauf als Returnwert zurückgibt. In der Funktion sind die `n` Namen und Matrikelnummern von der Standardeingabe einzulesen und im Feld einzutragen. Sie können voraussetzen, daß die Namen aus höchstens 20 Buchstaben bestehen.

Fehleingaben sind nicht abzufangen!

Aufgabe 3 (i2s003)

Was erscheint bei Ablauf des folgenden Programms, wenn von der Tastatur der Spruch

```
Man kommt nicht weit
wenn man nicht
von ganz vorne beginnt
#
```

eingegeben wird?

```
#include <stdlib.h>
#include <stdio.h>
#include <string.h>
typedef struct node{
    int l;
    char *s;
    struct node *e;
    struct node *n;} NODE;
```

```
NODE *ins( NODE *anker,char *word );

main()
{ char word[81];
  NODE *p,*q,*a=0;

  for (;;)
  {
    scanf("%s",word);
    if ( *word=='#' ) break;
    a = ins( a,word );
    word[50]=0;
  }

  for ( q=a ; q; q=q->n )
  { for ( p=q ; p; p=p->e ) printf("%s ", p->s);
    printf("\n");
  }
  printf("-----\n");
  for ( p=a ; p; p=p->n ) printf("%c ", *p->s);
  printf("\n");
  for ( p=a ; p; p=p->e ) printf("%c ", *p->s);
  printf("\n-----\n");
  for ( p=a ; p; )
  { printf("%c ", *p->s);
    if ( p->e ) p = p->e;
    else p = p->n;
  }
}
```

```
NODE *ins( NODE *anker,char *word )
{  NODE *q,*hilf,*p=anker;
   char *s;
   int len = strlen(word);
   s = (char*) malloc( len+1 );
   strcpy( s,word );
   hilf = (NODE*) malloc(sizeof(NODE));
   hilf->l = len;
   hilf->s = s;
   hilf->e = 0;

   q = 0;
   while ( p && p->l <= len )
   {  q = p;
      p = p->n;
   }
   hilf->n = p;

   if ( !q ) return hilf;

   if ( q->l == len )
   {
       while ( q->e )
       {  q->n = p;
          q = q->e;
       }
       q->e = hilf;
   }
   else
   do
   {  q->n = hilf;
      q = q->e;
   } while ( q );

   return anker;
}
```

Aufgabe 4 (i2s004)

An Ihrem PC sei die Parallelschnittstelle an der Adresse 0x378 vorhanden. An jedem der Pins 2 bis 9 ist ein Lämpchen angeschlossen, das leuchtet, wenn die Spannung „high“ am Pin anliegt. Mit dieser Parallelschnittstelle sollen Sie eine Lichtlaufkette mit folgenden Eigenschaften realisieren:

- Aus der Grundstellung, daß keine Lampe leuchtet, werden beginnend bei Pin 2 der Reihe nach die Lampen zum Leuchten gebracht, bis alle Lampen leuchten;
- nun werden wieder bei Pin 2 beginnend die Lampen der Reihe nach ausgeschaltet bis keine mehr leuchtet;
- zwischen zwei Schaltvorgängen liegt jeweils ein Takt von deltaT msec;

- nach `nsec` Sekunden bleibt die Kette in der dann erreichten Stellung stehen;

Schreiben Sie hierzu eine C-Funktion `kette`, der Sie die Parameter `deltaT` und `nsec` übergeben. Verwenden Sie in der C-Funktion eine Schleife, in deren Rumpf Sie das Datenbyte jeweils neu berechnen bevor Sie es in das Datenregister eintragen.

Hinweis: Verwenden Sie für die Verzögerung die Funktion `delay()` und vernachlässigen Sie die Laufzeit der sonstigen Befehle.

```

Pin 9:-----#####-----#####-----#####-----#
Pin 8:-----#####-----#####-----#####-----##
Pin 7:-----#####-----#####-----#####-----###
Pin 6:-----#####-----#####-----#####-----####
Pin 5:----#####-----#####-----#####-----#####
Pin 4:---#####-----#####-----#####-----#####
Pin 3:--#####-----#####-----#####-----#####
Pin 2:-#####-----#####-----#####-----#####

```

Bedeutungen:

- : Lampe leuchtet nicht (Spannung low)
- # : Lampe leuchtet (Spannung high)

Jede Spalten ist ein neuer Takt der Laenge `deltaT` Millisekunden

Aufgabe 5 (i2s005)

Schreiben Sie ein C-Programm, das Ihren Namen in eine Textdatei schreibt. Der Name dieser Datei sei wie folgt festgelegt:

- Wird das Programm von der Kommandozeile mit genau einem Argument gestartet, so ist dieses Argument der Dateiname;
- Wird das Programm ohne Argument gestartet, so wird nachgeschaut, ob eine Systemvariable `DATEI` gesetzt ist, deren Inhalt dann als Name genommen wird. Ist eine solche Systemvariable nicht gesetzt, so wird die Datei `default.txt` genannt;
- Wird das Programm mit zwei oder mehr Argumenten gestartet, so wird es vorzeitig nach Ausgabe einer Fehlermeldung abgebrochen;