

|   |                 |
|---|-----------------|
| Hochschule Deggendorf<br>Dr. Peter Jüttner  |                 |
| Vorlesung: Einführung in die Programmierung | WS 2013         |
| Übung 7                                     | Termin 21.11.13 |

## Zeichenketten und zusammengesetzte Datentypen - Musterlösung

### 1. Zeichenketten

Schreiben Sie zwei Versionen eines Programms, das zwei Zeichenketten einliest, aneinanderfügt und die zusammengesetzte Zeichenkette wieder ausgibt. Nutzen Sie für die erste Version keine Bibliotheksroutinen außer zur Ein-/Ausgabe. Für die zweite Version gilt diese Einschränkung nicht.

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

```
int main(void)
{ char s1[101]; /* 1. Zeichenkette */
  char s2[101]; /* 2. Zeichenkette */
  int i, j;
```

```
    printf("Zusammenfuegen zweier Zeichenketten\n");
    printf("1.Version ohne strcat \n");
```

```
    printf("Bitte 1. Zeichenkette eingeben (max 50 Zeichen)\n");
    gets(s1); /* alternativ scanf("%s",s); scanf liest ggf. nur bis zum 1.
Leerzeichen*/
```

```
    printf("Bitte 2. Zeichenkette eingeben (max 50 Zeichen)\n");
    gets(s2); /* alternativ scanf("%s",s); scanf liest ggf. nur bis zum 1.
Leerzeichen*/
```

```
    /* Suchen des Endezeichens im 1. String s1 */
    i = 0;
```

```

while ((i<101) && (s1[i] != '\0'))
    i++;

/* zeichenweises Anfügen des zweiten Strings s2 an s1 bis Endezeichen
gefunden*/
j=0;
while((s2[j] != '\0') && ((i+j)<100))
{ s1[i+j] = s2[j];
  j++;
};

/* Anfügen Endezeichen an s1 */
s1[i+j] = '\0';

/* Ausgabe der aneinandergefügten Zeichenketten */
printf("aneinandergefuegte Zeichenketten:\n");
printf("%s\n",s1);

printf("Zusammenfuegen zweier Zeichenketten\n");
printf("2.Version mit strcat \n");

printf("Bitte 1. Zeichenkette eingeben (max 50 Zeichen)\n");
gets(s1); /* alternativ scanf("%s",s); scanf liest ggf. nur bis zum 1.
Leerzeichen*/

printf("Bitte 2. Zeichenkette eingeben (max 50 Zeichen)\n");
gets(s2); /* alternativ scanf("%s",s); scanf liest ggf. nur bis zum 1.
Leerzeichen*/

/* Sicherstellen, dass beide Zeichenketten in s1 Platz haben */
if (strlen(s1)+strlen(s2) <= 100)
{ strcat(s1,s2);
  printf("aneinandergefuegte Zeichenketten:\n");
  printf("%s\n",s1);
}
else printf("Zeichenketten können nicht konkateniert werden, da zu lang\n");

system("PAUSE");
return 0;
}

```

## 2. Zusammengesetzte Datentypen

Definieren Sie in C einen zusammengesetzten Datentyp Kfz, der folgende Komponenten hat:

- Art, z.B. Pkw, Lkw, Motorrad

- Typ, z.B. Audi A6, VW Golf, BMW R75
- Kilometerstand
- Kennzeichen

Wählen Sie für die Komponenten geeignete Typen. Schreiben Sie ein Programm, das für ein Kfz die Daten einliest und wieder ausgibt.

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

int main(void)
{
    typedef enum Kfz_art { Pkw, Lkw, Mot };

    typedef struct Kfz
    {
        Kfz_art art;
        char typ[40];
        char kennz[11]; /* max. 3Buchst. - max. 2 Buchst max 4 Ziffern */
    };

    Kfz fahrzeug;
    int art = -1;

    /* Einlesen der Fahrzeugdaten */
    /* 1. Einlesen Art */
    while ((0 > art) || (art > 2))
    { printf("Bitte Art eingeben (0 = Pkw, 1 = Lkw, 2 = Mot)\n");
      scanf("%d",&art);
      if ((0 > art) || (art > 2))
          printf("falsche Eingabe\n");
      else fahrzeug.art = (Kfz_art) art;
    };
    /* 2. Einlesen Typ */
    printf("Bitte Typ eingeben (max. 40 Zeichen)\n");

    /* Lösche Eingabepuffer wegen gemischter Verwendung von scanf und
    gets */
    fflush(stdin);
    gets(fahrzeug.typ);

    /* 3. Einlesen Kennzeichen */
    printf("Bitte Kennzeichen in der Form BBB-BB-ZZZZ eingeben\n");
    gets(fahrzeug.kennz);

    /* Ausgabe Fahrzeugdaten */
```

```
printf("fahrzeugart:");  
if (fahrzeug.art == Pkw)  
    printf("Pkw\n");  
else if (fahrzeug.art == Lkw)  
    printf("Lkw\n");  
else printf("Motorrad\n");  
printf("Typ:");  
printf("%s\n", fahrzeug.typ);  
printf("Kennzeichen:");  
printf("%s\n", fahrzeug.kennz);  
  
system("PAUSE");  
return 0;  
}
```