

Lösung Übungszettel 6

1 Aufgabe 5

1.1 Die Bibliothek *biblio4.c*

```
/* Bibliothek zum Addieren und Subtrahieren von
   Integer- und Double-Zahlen und Zeichenketten
   Allgemeine Version mit void-Zeigern
*/

//Einbinden von <string.h> fuer die Bearbeitung von Zeichenketten
#include <string.h>

//Deklariere von globalen Variablen für die Ergebnisse
//lokale Variablen werden gelöscht, wenn eine Funktion
//abgearbeitet ist, deshalb zeigt ein Zeiger auf eine
//lokale Variable dann ins Nichts
double ergebnis1;
int ergebnis2;
char *ergebnis3;

//Funktion zum Addieren
//typ zum Feststellen des Types
//('i' = Integer, 'f' = Double, 's' = String)
//a ist Zeiger auf ersten Summand
//b ist Zeiger auf zweiten Summand
//Rückgabewert: Zeiger auf Ergebnis
void *add(char typ, void *a, void *b)
{
    //Variable fuer die Zeichenlaenge der Teilstrings
    int zeichen;

    switch(typ)
    {
        case 'i':
            //1. Fall: Integer
            //Ergebnis berechnen, Casting für Integer aus void!
            ergebnis2 = *(int *)a + *(int *)b;
            //Ergebnis zurueckgeben
            return &ergebnis2;
        case 'f':
            //2. Fall: Double
```

```

        //Ergebnis berechnen, Casting für Double aus void!
        ergebnis1 = *(double *)a + *(double *)b;
        //Ergebnis zurueckgeben
        return &ergebnis1;
    case 's':
        //3. Fall: String
        //Laenge des Ergebnisstrings berechnen:
        //Laenge des ersten Teilstrings +
        //Laenge des zweiten Teilstrings + 1 für das Endzeichen \0
        zeichen = strlen((char *)a) + strlen((char *)b);
        //Speicher fuer den Ergebnisstring bereitstellen
        ergebnis3 = (char *)malloc(zeichen * sizeof(char) -1);
        //Ersten Teilstring in den Ergebnisstring kopieren
        strcpy(ergebnis3, (char *) a);
        //Zweiten Teilstring anhaengen
        strcat(ergebnis3, (char *) b);
        //Ergebnis zurueckgeben
        return ergebnis3;
    }
}

```

```

//Funktion zum Subtrahieren
//typ zum Feststellen des Types
//('i' = Integer, 'f' = Double, 's' = String)
//a ist Zeiger auf Minuend
//b ist Zeiger auf Subtrahend
//Rückgabewert: Zeiger auf Ergebnis
void *sub(char typ, void *a, void *b)
{
    //Variable, um Position in einem String zu merken
    char * aktuell;
    //Variable, um Startposition eines Teilstrings zu merken
    char * start;
    //temporaere Variablen
    char *temp,*remember;
    //Variable fuer die Zeichenlaenge der Teilstrings
    int zeichen;
    //Variable fuer die Schleife
    int i;

    switch(typ)
    {
        case 'i':
            //1. Fall: Integer
            //Ergebnis berechnen, Casting für Integer aus void!
            ergebnis2 = *(int *)a - *(int *)b;
            //Ergebnis zurueckgeben
            return &ergebnis2;
        case 'f':
            //2. Fall: Double
            //Ergebnis berechnen, Casting für Double aus void!

```

```

    ergebnis1 = *(double *)a - *(double *)b;
    //Ergebnis zurueckgeben
    return &ergebnis1;
case 's':
    //3. Fall: String
    //Laenge des maximalen Ergebnisstrings berechnen:
    //Laenge des ersten Teilstrings +
    //Laenge des zweiten Teilstrings + 1 für das Endzeichen \0
    zeichen = strlen((char *)a) + strlen((char *)b) + 1;
    //Speicher fuer den Ergebnisstring bereitstellen
    ergebnis3 = (char *)malloc(zeichen * sizeof(char) + 1);
    //Pruefen, ob der zweite String im ersten enthalten ist
    if((start = strstr((char *)a, (char *)b)) == NULL)
        //String ist nicht enthalten
        //Ergebnis zurueckgeben
        return (void *)a;
else
{
    //wenn der String mehrmals geloescht wird, aendert
    //sich das Wort, deshalb Merkvariable (a soll nicht
    //veraendert werden
    remember = (char *)malloc(zeichen * sizeof(char) + 1);
    //a nach remember kopieren
    strcpy(remember, a);
    //ansonsten alle Vorkommen loeschen
    while((start = strstr((char *)remember, (char *)b)) != NULL)
    {
        //String ist enthalten
        //Aktuell wird zum Kopieren der Zeichen auf den Anfang
        //des Ergebnisstrings gesetzt
        aktuell = ergebnis3;
        //temp ist der aktuelle Buchstabe in dem Wort, aus dem
        //geloescht wird
        temp = remember;
        //Zeichen vom Anfang der ersten Zeichenkette bis zum
        //Beginn des zu loeschenden Teilstrings kopieren
        while((char *)temp < start)
        {
            *aktuell++ = *((char *)temp);
            temp++;
        }
        //den Teilstring ueberspringen, diese Zeichen sollen
        //geloescht werden
        for(i = 0; i < strlen((char *)b); i++)
            temp++;
        //den Rest der ersten Zeichenkette in den Ergebnisstring
        //kopieren
        while(*(char *)temp)
        {
            *aktuell++ = *((char *)temp);
            temp++;
        }
    }
}

```

```

    }
    //das neue Wort, aus dem eventuell geloescht werden soll
    //nach remember kopieren
    strcpy(remember, ergebnis3);
    }
    //Ergebnis zurueckgeben
    return (void *)ergebnis3;
}
}
}

```

1.2 Die Header-Datei *biblio4.h*

```

/* Header-Datei zur Bibliothek biblio4.c */

//Funktionsprototypen definieren
void *add(char typ, void *a, void *b);
void *sub(char typ, void *a, void *b);

```

1.3 Das Programm *addieren4.c*

```

/* Anwendung der Bibliothek biblio4.c */

//<stdio.h> einbinden für Ausgabe mit printf
// "biblio4.h" einbinden für die Nutzung von biblio3.c
#include <stdio.h>
#include "biblio4.h"

int main()
{
    //Double- und Integer-Zahlen deklarieren und initialisieren
    double a = 7.3, b = 3.891;
    int c = 11, d = 58;
    //Zwei Strings zum Addieren deklarieren und initialisieren
    char string1[7] = "Hello ";
    char string2[6] = "World";
    //Teilstring zum Subtrahieren deklarieren und initialisieren
    char string3[3] = "l";

    //Die Funktionen der Bibliothek benutzen und Ergebnisse ausgeben
    printf("%g + %g = %g\n", a, b, *(double *)add('f', &a, &b));
    printf("%g - %g = %g\n", a, b, *(double *)sub('f', &a, &b));
    printf("%d + %d = %d\n", c, d, *(int *)add('i', &c, &d));
    printf("%d - %d = %d\n", c, d, *(int *)sub('i', &c, &d));
    printf("%s + %s = %s\n", string1, string2, (char *)add('s', string1, string2));
    printf("%s - %s = %s\n", string1, string3, (char *)sub('s', string1, string3));
}

```