

C01 Kleine Programmieraufgaben in C

panitz

Zusammenfassung

In diesem Kurs befinden sich einfache kleine Aufgaben zur Programmierung in der Programmiersprache C.

Frage: Map auf Zahlenarray

Schreiben Sie eine Funktion, die eine übergebene Funktion auf alle Elemente eines Arrays anwendet und die Arrayelemente mit dem Funktionsergebnis ersetzt.

C

```
#include "map.h"
void map(int f(int), unsigned int argc, int xs []){
}
```

C

C

```
#include "map.h"
void map(int f(int), unsigned int argc, int xs []){
    int i;
    for (i=0; i<argc; i++){
        xs[i] = f(xs[i]);
    }
}
```

C

Erläuterung

Antworttext

Frage: Quersumme

Schreiben Sie eine Funktion, die von einer Zahl so lange die Quersumme berechnet, bis Sie eine einstellige Zahl erhalten.

C

```
#include "quer.h"

unsigned long int quer(unsigned int x){
    return 0;
}
```

C

C

```
    unsigned long int quer(unsigned int x){  
    if (x<10) return x;  
    return quer(x%10+quer(x/10));  
}
```

C

Erläuterung

Mit der Modulorechnung zu

Frage: Test auf Stringpräfix

Schreiben Sie eine Funktion, die testet, ob ein String Präfix eines zweiten Strings ist. Die einfachste Lösung hierzu ist die rekursive Lösung.

C

```
#include "pref.h"
bool isPrefix(char* pref, char* str){
    return false;
}
```

C

C

```
#include "pref.h"
bool isPrefix(char* pref, char* str){
    if (*pref == '\0') return true;
    if (*pref != *str) return false;
    return isPrefix(++pref, ++str);
}
```

C

Erläuterung

In der Lösung wird erst getestet, ob der pref der leere String ist. Dann ist er Präfix. Anschließend wird geschaut, ob beide String mit einem unterschiedlichen Zeichen beginnen. Dann kann es kein Präfix sein. Dieses schließt den Fall ein, dass der zweite String ein leerer String ist. Für die Rekursion werden schließlich beide String ein Zeichen weiter gesetzt.

Frage: String in umgekehrter Reihenfolge

Schreiben Sie eine Funktion, die einen neuen String erzeugt, der die Zeichen des Arguments in umgekehrter Reihenfolge enthält.

C

```
#include "rev.h"
#include <stdlib.h>

char* rev(char* str){
    int laenge = strlen(str);

    return NULL;
}
```

C

C

```
#include "rev.h"
#include <stdlib.h>

char* rev(char* str){
    int laenge = strlen(str);
    char* result = malloc(laenge+1);
    result[laenge]='\0';
    int i;
    for (i=0;i<laenge;i++) result[i]=str[laenge-i-1];
    return result;
}
```

C

Erläuterung

Antworttext

Frage: Test

Schreiben Sie eine Funktion, die testet, ob ein Array aufsteigend sortiert ist.

C

```
#include "sorted.h"
bool isSorted(unsigned int argc, int* xs){
    return false;
}
```

C

C

```
#include "sorted.h"
bool isSorted(unsigned int argc, int* xs){
    if (argc==0) return true;
    int i;
    for (i=0;i<argc-1;i++)
        if (xs[i]>xs[i+1]) return false;
    return true;
}
```

C

Erläuterung

Es werden die nebeneinander liegenden Elemente verglichen. Sind diese in falscher Reihenfolge, so ist der Array nicht sortiert. Eine kleine Falle sind leer Arrays. Hier ist zu beachten, dass $0-1 > 0$ ist, weil der Typ `unsigned int` keine negativen Zahlen kennt.

Frage: Test auf Pallindrom

Schreiben Sie eine Funktion, die testet, ob der übergebene String ein Pallindrom ist.

C

```
#include "pal.h"
bool isPal(char* str){
    int laenge = strlen(str);
    return false;
}
```

C

C

```
#include "pal.h"
bool isPal(char* str){
    int laenge = strlen(str);
    int i;
    for (i=0;i<laenge / 2;i++)
        if (str[i]!=str[laenge-i-1]) return false;
    return true;
}
```

C

Erläuterung

Antworttext

Frage: Summe in eine Array

Berechnen Sie die Summe der Zahlen in einer Reihung

C

```
#include "sum.h"

long int sum(unsigned int argc,int* args){
    return 0;
}
```

C

C

```
#include "sum.h"

long int sum(unsigned int argc,int* args){
    unsigned int result = 0;
    int i;
    for (i=0;i<argc;i++)result+=args[i];
    return result;
}
```

C

Erläuterung

Es ist einmal durch alle Elemente zu iterieren.

Frage: Anzahl eines Zeichens im String

Schreiben Sie eine Funktion, die angibt, wie oft ein bestimmtes Zeichen in einem String enthalten ist.

C

```
#include "countChar.h"

unsigned int countChar(char c, char* str){
    unsigned int result = 0;
    return result;
}
```

C

C

```
    #include "countChar.h"

    unsigned int countChar(char c,char* str){
        unsigned int result = 0;
        for (;*str != '\0';str++) if (*str==c)result++;
        return result;
    }
```

C

Erläuterung

Es ist daran zu denken, dass Strings 0-terminiert sind. So lässt sich einfach durch einen String iterieren.

Frage: Fakultät

Schreiben Sie die Fakultätsfunktion.

C

```
    #include "factorial.h"

    unsigned long int factorial(unsigned int n){
        return 0;
    }
```

C

C

```
#include "factorial.h"

unsigned long int factorial(unsigned int n){
    return n==0?1:n*factorial(n-1);
}
```

C

Erläuterung

Man beachte die Typen. Wir rechnen nur auf positiven Zahlen. Der Rückgabotyp hat 64 Bits, der Argumenttyp hingegen nur 32 Bits.