
Praktikum 4 Abgabe bis 03.01.2021

Name, Vorname: Özkurt, Cihat

Mat.-Nr.: 11148632

GMID: inf2323

Datum: 03.01.2021

Betreuer im Praktikum: Gross, Julien

Aufgabe 1: Programmieren in C

- Schreiben Sie eine Funktion `int sucheZeichen(const char s[], char c)`, wobei der Rückgabewert der Häufigkeit des Characters `c` im Char-Array `s` entspricht.
- Schreiben Sie eine Funktion `int laengeZeichenkette(const char s[])`, die die Anzahl der Zeichen (die Länge der Zeichenkette) als Rückgabewert liefert.
- Schreiben Sie dann ein Hauptprogramm, das eine kleine Menüsteuerung enthält, um nach einem Buchstaben in einem Namen zu suchen.

Der Ablauf soll dann so aussehen:

Bitte Namen eingeben: Isabella

Isabella hat 8 Zeichen.

Möchtest Du nach einem Buchstaben in Isabella suchen (j / n)? j

Bitte Buchstaben eingeben: a

Der Buchstabe a kommt 2-mal vor.

Möchtest Du nach einem Buchstaben in Isabella suchen (j / n)? j

Bitte Buchstaben eingeben: k

Der Buchstabe k kommt 0-mal vor.

Möchtest Du nach einem Buchstaben in Isabella suchen (j / n)? n

Auf Wiedersehen!

Hinweise:

- Verwenden Sie aus Übungsgründen im Menü die `switch` Anweisung, die in den Case Labels die Character `j` und `n` auswertet.
- Verwenden Sie `fflush(stdin)` aus der Bibliothek `<stdio.h>`, um Probleme mit dem Tastaturpuffer bei der Eingabe zu umgehen.

Lösung:

<Kopieren Sie bitte Ihr Programm hier hin. >

```

#include <stdio.h>

int sucheZeichen(const char s[], char c){
    int count=0;
    int i=0;
    while(s[i] != '\0'){
        if(s[i] == c)
            count++;
        i++;
    }
    return count;
}

int laengeZeichenkette(const char s[]){
    int count=0;
    int i=0;
    while(s[i] != '\0'){
        count++;
        i++;
    }
    return count;
}

int main(){

    const char a[50];
    char b;
    int flagge=1;

    printf("Bitte Namen eingeben:\n");
    scanf("%s",&a);
    printf("%s hat %i Zeichen\n",a,laengeZeichenkette(a));

    while(flagge==1){

        printf("Möchtest Du nach einem Buchstaben in %s suchen (j / n)?\n",a);
        char eingabe;
        int c;
        while ((c = getchar()) != EOF && c != '\n');
        scanf("%c",&eingabe);

        switch (eingabe){

            case 'j':
                printf("Bitte Buchstaben eingeben:\n");
                while ((c = getchar()) != EOF && c != '\n');
                scanf("%c",&b);
                printf("Der Buchstaben %c kommt %i mal vor\n",b,sucheZeichen(a,b));
                break;

            case 'n':

```

```

        printf("Auf wiedersehen");
        flagge=0;
        break;
    }
}

return 0;
}

```

Aufgabe 2: Programmieren in C

Organisieren Sie den Code aus Aufgabe 1 um. Verwenden Sie Funktionsdeklarationen für `sucheZeichen(...)` und `laengeZeichenkette(...)` und schreiben Sie diese in die Headerdatei `myfunctions.h`. Verschieben Sie den Code der beiden Funktionen in die Datei `myfunctions.c`. Verschieben Sie das Mainprogramm in die Datei `mymain.c`. Schreiben Sie ein make-File `mymake.mak`, das das Programm `myprogram` generiert.

Lösung:

<Kopieren Sie bitte den Inhalt Ihrer Dateien hier hin und schreiben Sie den Dateinamen jeweils über den entsprechenden Code. >

Die Datei: `myfunctions.h`

```

int sucheZeichen(const char s[], char c);
int laengeZeichenkette(const char s[]);

```

Die Datei: `myfunctions.c`

```

#include "myfunctions.h"

```

```

int sucheZeichen(const char s[], char c){
    int count=0;
    int i=0;
    while(s[i] != '\0'){
        if(s[i] == c)
            count++;
        i++;
    }
    return count;
}

int laengeZeichenkette(const char s[]){
    int count=0;
    int i=0;
    while(s[i] != '\0'){
        count++;
        i++;
    }
}

```

```

}
return count;
}

```

Die Datei: mymain.c

```

#include <stdio.h>
#include "myfunctions.h"

int main(){

    const char a[50];
    char b;
    int flagge=1;

    printf("Bitte Namen eingeben:\n");
    scanf("%s",&a);
    printf("%s hat %i Zeichen\n",a,laengeZeichenkette(a));

    while(flagge==1){

        printf("Möchtest Du nach einem Buchstaben in %s suchen (j / n)?\n",a);
        char eingabe;
        int c;
        while ((c = getchar()) != EOF && c != '\n');
        scanf("%c",&eingabe);

        switch (eingabe){

            case 'j':
                printf("Bitte Buchstaben eingeben:\n");
                while ((c = getchar()) != EOF && c != '\n');
                scanf("%c",&b);
                printf("Der Buchstaben %c kommt %i mal vor\n",b,sucheZeichen(a,b));
                break;

            case 'n':
                printf("Auf wiedersehen");
                flagge=0;
                break;

        }
    }

    return 0;
}

```

Die Datei: mymake.mak

```

myprogram: myfunctions.o mymain.o
    gcc -o myprogramm myfunctions.o mymain.o

myfunctions.o: myfunctions.h myfunctions.c
    gcc -o myfunctions.c

mymain.o: myfunctions.h mymain.c
    gcc -c mymain.c

```

Aufgabe 3: Programmieren in C

Schreiben Sie ein Programm zur Lagerverwaltung. Wir wollen hierzu ein **struct** verwenden, das die folgenden Komponenten hat:

```
char artikel[laengeArtikel];  
int anzahl;
```

Das Array **lager[anzahlEintraege]** enthält dann als Array-Elemente die Informationen zu Artikeln gemäß der obigen Struktur.

Folgende Funktionen sollen geschrieben werden:

- Artikel hinzufügen.
Ein Artikel mit Anzahl soll in die Lagerliste eingetragen werden, falls noch Platz ist und falls der Artikel noch nicht existiert. Falls der Artikel bereits vorhanden ist, soll die Anzahl entsprechend erhöht werden.
- Artikel entnehmen.
Geben Sie einen Artikel ein und geben Sie an, wie viele Exemplare aus dem Lager entnommen werden. Aktualisieren Sie die Anzahl. Beachten Sie den Fall, dass der Artikel nicht existiert, und den Fall, wo nicht genügend Exemplare auf Lager sind. Erzeugen Sie dann entsprechende Meldungen für den Anwender.
- Eintrag suchen.
Für einen Artikel soll ausgegeben werden, welche Anzahl noch im Lager ist. Formulieren Sie auch hier alle möglichen Meldungen an den Anwender.
- Lager als Tabelle ausgeben.
Hier soll eine Tabelle mit den Spalten Artikel und Anzahl ausgegeben werden. Am Ende der Tabelle sollen die Anzahlen addiert werden, damit man weiß, wie viele Artikel insgesamt im Lager sind.
- Schreiben Sie eine Menüsteuerung, durch die sich die einzelnen Funktionen aufrufen lassen.

Hinweise:

- Sie werden in der Aufgabe Zeichenketten vergleichen müssen. Verwenden Sie hierzu die Bibliotheksfunktion `int strcmp(const char *s1, const char *s2)`.
- Schreiben Sie alle Funktionen in das **main**-Programm. Sie sollen für dieses Beispiel kein **make** und keine separate Übersetzung verwenden.

Lösung:

<Kopieren Sie bitte Ihr Programm hier hin. >

```

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

typedef struct{
    char artikel[10];
    int anzahl;
}Lager;

int anzahlEintraege = 5; //anzahlEintraege ist Kapazität von Lager List also
//Können wir höchstens 5 Mal produkt eingeben
int count=0; //count zeigt uns wie viele struct(object) schon haben wir und
//index für nächste Artikel

void hinzufügen(Lager *lager,Lager x){
    if(count < anzahlEintraege) // count muss niedriger als unserer Kapazität
    //um eine neue Produkt einzugeben
    if (suche(lager,x.artikel) == -1){ // Falls eingegebene Artikel nicht in
    //Lager List ist
        lager[count] = x; // einfach hinzufügen in count index
        count++;
    }
    else{
        for (int i = 0; i < count; i++){
            if(strcmp(x.artikel, lager[i].artikel) == 0 ) // Falls eingegebene
            //Artikel in Lager List ist
                lager[i].anzahl++; // wir hinfügen nur Anzahl von Artikel z.B in
            //1.index haben wir 1 Mal iphone jetzt erhöhen wir nur iphone Zahl.Wir addieren
            //keine neue Artikel
        }
    }
    else
        printf("es gibt noch kein Platz\n"); // Falls (count < anzahlEintraege)
        //ist falsch das bedeutet unsere Kapazität ist voll.es gibt noch kein Platz
}

// Kontroll von eingegebene Produkt ob er ist schon in Lager List
int suche(Lager *p, char artikel[]){
    for (int i = 0; i < count; i++){
        if(strcmp(p[i].artikel,artikel) == 0 ) //Wir vergleichen artikel Namen in
        //Lager List mit eingegebene artikel
            return p[i].anzahl; //falls eingegebene Artikel schon drin ist
    }
    return -1; //falls eingegebene Artikel nicht drin ist
}

void entnehmen(Lager *lager,Lager x, int anzahl){
    int temp = suche(lager,x.artikel); // temp speichern Anzahl von gegeben
    //Artikel
    if(temp != -1 ){ //temp = -1 beduetet eingegebene Artikel ist nicht in
    //Lager List

```

```

        if(temp >= anzahl){ // Falls unsere Anzahl von Artikel gleich oder mehr
als Anzahl von gegene Artikel Anzahl z.B wir haben 3 Mal Iphone wollen wir 1
Mal Iphone entnehmen
            for (int i = 0; i < count; i++){
                if(strcmp(x.artikel, lager[i].artikel) == 0 ){ // suchen wir index
von List sprechend eingegebene Artikel Nahme
                    lager[i].anzahl -= anzahl; // reduzieren Anzahl von Artikeln
                }
            }

        }
        else
            printf("Es gibt weniger Artikel im Lager als gegebene Anzahl!\n");
    }
    else
        printf("Der Artikel existiert nicht!\n");
}

void tabelle(Lager *lager){
    printf("Artikel\t|\tAnzahl\n");
    int summe = 0; //summe -> summe Anzahl von alle Artikel
    for (int i = 0; i < count; i++){
        printf("%s\t\t|\t\t%i\n", lager[i].artikel, lager[i].anzahl);
        summe += lager[i].anzahl;
    }
    printf("Die Summe der Artikel %d\n", summe);
}

int main(void) {

    int n;
    Lager lager[5];

    int flag=1;
    printf("Menu\n");
    printf("1)Artikel hinzufügen.\n");
    printf("2)Artikel enthnehmen.\n");
    printf("3)Eintrag suchen.\n");
    printf("4)Artikel Lager als Tabelle ausgeben..\n");
    printf("5)Ende des Programrams.\n");

    while(flag==1){
        Lager newProdukt;

        printf("Wahlen eine eingabe 1,2,3,4,5\n");
        scanf("%i",&n);

        if(n==1){

            printf("Artikel Name\n");

```

```

        scanf("%s", &newProdukt.artikel);
        printf("Artikel Zahl\n");
        scanf("%i", &newProdukt.anzahl);
        hinzufügen(lager,newProdukt);
    }

    else if(n==2){
        printf("Artikel Name\n");
        scanf("%s", &newProdukt.artikel);
        printf("Artikel Zahl\n");
        scanf("%i", &newProdukt.anzahl);
        entnehmen(lager,newProdukt, newProdukt.anzahl);
    }

    else if(n==3){

        printf("Artikel Name\n");
        scanf("%s", &newProdukt.artikel);
        printf("Es gibt insgesamt %i Mal Artikel von %s\n",suche(lager,
newProdukt.artikel),newProdukt.artikel);
    }
    else if(n==4)
        tabelle(lager);
    else if(n==5)
        flag=0;
    }
    return 0;
}

```


Aufgabe 4: Programmieren in C

Sie kaufen ein Auto und stellen fest, dass im Fahrzeugschein das Datum der Erstzulassung 13.06.2008 steht. Schreiben Sie ein C-Programm, das für ein beliebiges Zulassungsdatum, das über die Tastatur eingegeben wird, das Alter eines Autos berechnet.

Die Ausgabe des Programms sollte in etwa so aussehen:

Programm zur Berechnung des Alters eines Autos

Bitte geben Sie das Zulassungsdatum ein (in der Form tt.mm.jjjj): _____

Das Auto ist _____ Jahre alt.

Implementieren Sie dieses Programm mit den folgenden Funktionen:

- (1) **int eingabeZulassungsdatum()**
Dies erfordert etwas Überlegung: Es soll von der Tastatur das Zulassungsdatum in der Form tt.mm.jjjj eingegeben werden. Daraus wird das Zulassungsjahr jjjj (als int) extrahiert. Dieses wird als Returnwert zurückgegeben.
- (2) **int alterBerechnung(int zulassungsJahr)**
Das Zulassungsjahr wird als Parameter übergeben, das Alter wird berechnet und als Returnwert zurückgegeben.
- (3) **void ausgabeAlter(int alter)**
Das Alter wird als Parameter übergeben und die obige Ausgabe wird erzeugt.
- (4) **int main()**
Hier werden die obigen Funktionen aufgerufen.

Lösung:

<Kopieren Sie bitte Ihr Programm hier hin. >

```

#include <stdio.h>
int eingabeZulassungsdatum(){

    int tag,monat,jahr;
    printf("Bitte geben Sie das Zulassungsdatum ein (in der Form
tt.mm.jjjj):\n");
    scanf("%i.%i.%i", &tag, &monat, &jahr);
    return jahr;
}
int alterBerechnung(int zulassungsJahr){
    int alt;
    alt=2021 - zulassungsJahr;
    return alt;
}
void ausgabeAlter(int alter){
    printf("Das Auto ist %i Jahre alt",alter);
}

int main(){
    printf("Programm zur Berechnung des Alters eines Autos\n");
    ausgabeAlter(alterBerechnung(eingabeZulassungsdatum()));

    return 0;
}

```

Aufgabe 5: Programmieren in C

Schreiben Sie ein C-Programm, das die Funktion `floor` aus der mathematischen C-Bibliothek benutzt:

```
double floor(double x)      The C library function double floor(double x)
                             returns the largest integer value less than or
                             equal to x.
```

Die Ausgabe des Programms sollte in etwa so aussehen:

Programm zur Floorfunktion

Bitte geben Sie den Wert ein: 3.14 (als Beispiel)

Der Floorwert von 3.140000 ist: 3.000000

Hinweise:

- Machen Sie sich klar, was die `floor`-Funktion (Gaußklammer) macht. Wie lautet der Floorwert von -3.14?
- Sie benutzen die C-Bibliothek für mathematische Funktionen. Zum einen benötigen Sie daher das Headerfile `<math.h>` und zum anderen müssen Sie natürlich die Bibliothek bei der Übersetzung mit `-lm` hinzu linkern.

Lösung:

<Kopieren Sie bitte Ihr Programm hier hin. >

```
#include <stdio.h>
#include <math.h> // Bibliothek von mathe funktionen

double convert(double a){
    a = floor( a); // umrechnun von Floorwert
    return a;
}

int main(){
    double a; // eingegebene zahl

    printf("Programm zur Floorfunktion\n");
    printf("Bitte geben Sie den Wert ein: 3.14 (als Beispiel)\n");
    scanf("%lf",&a);

    printf("Der Floorwert von %lf ist: %lf",a,convert(a));
    return 0;
}
```

Aufgabe 6: Programmieren in C

Schreiben Sie ein Programm, das einige Dinge mit Zeichenketten (Strings) macht. Formulieren Sie alle auszuführenden Aktionen als Funktionen und speichern Sie diese in einem separaten File `stringFunctions.c` ab. Schreiben Sie die Funktionsdeklarationen in das Header-File `stringFunctions.h` und führen Sie die Übersetzung des Programms mit `make` aus.

Das Hauptprogramm (in `stringMain.c`) soll folgende Aktionen in einem Menü anbieten:

- Einlesen von zwei Firmennamen.
Legen Sie hierzu zwei `char`-arrays an.
- Vergleich der beiden Firmennamen auf Gleichheit.
Verwenden Sie die (selbst geschriebene) Funktion `int equal(char x[], char y[])`, wobei der Rückgabewert 1 ist, falls Gleichheit gilt, sonst 0.
- Nur wenn die beiden Firmennamen ungleich sind, sollen diese zu einem String verbunden werden. Verwenden Sie hierzu eine (selbst geschriebene) Funktion `stringconcatenate`, die zwei Strings als Eingabeparameter hat und einen dritten String erzeugt, der die beiden Strings durch ein `&`-Zeichen verbunden enthält.
- Dann soll im Hauptprogramm folgendes ausgegeben werden:
`<name_1> & <name_2> sind Konkurrenten.` oder
`<name_1> hat keine Konkurrenten.`

Die Ausgabe des Programms sollte in etwa so aussehen:

Programm zur Konkurrenzermittlung

Bitte geben Sie den 1. Firmennamen ein: Apple (als Beispiel)

Bitte geben Sie den 2. Firmennamen ein: Microsoft (als Beispiel)

Status: Apple & Microsoft sind Konkurrenten.

oder:

Bitte geben Sie den 1. Firmennamen ein: Apple

Bitte geben Sie den 2. Firmennamen ein: Apple

Status: Apple hat keine Konkurrenten.

Lösung:

<Kopieren Sie bitte Ihr Programm hier hin. >

```

#include <stdio.h>

int equal(char x[], char y[]){
int i=0;
while(x[i] != '\0' && y[i] != '\0'){ // ich kontrolliere jede index von
beide arrays(Buchstaben)
if(x[i] != y[i]){ // falls eine index Wert(Buchstaben) ist
unterschiedlich,das beduetet beide arrys sind nicht gleich
return 0;
}
i++;
}
return 1; //falls alle
}

void concatenate(char x[], char y[]) {
int i, j;

i = 0;

while (x[i] != '\0') { //erhöhe ich index bis ende von x array
i++;
}
// danach addiere ich ein Mal Leer,& und noch ein Mal leere
x[i] = ' ';
i++;
x[i] = '&';
i++;
x[i] = ' ';
i++;
j = 0; // index von y array
while (y[j] != '\0') { //erhöhe ich index bis ende von y array
x[i] = y[j]; //assignment y array in x array also beide array sind in x
array
j++;
i++;
}
x[i] = '\0';

printf("%s sind Konkurrenten",x);

}

void start(){
char x[50];
char y[50];
printf("Bitte geben Sie den 1. Firmennamen ein: Apple (als Beispiel)\n");
scanf("%s",x);
printf("Bitte geben Sie den 2. Firmennamen ein: Microsoft (als
Beispiel)\n");
scanf("%s",y);

```

```
if(equal(x,y) == 0)
    concatenate(x, y);
else
    printf("%s hat keine Konkurrenten",x);
}
int main(){
    start();

return 0;
}
```

Merry Christmas and Happy New Year 2021 !

