

# Praktische Aufgaben

## Fehlersuche und Debugging in C

### Dokumentinformationen

Dokumenttitel: Praktische Aufgaben  
Thema: Fehlersuche und Debugging in C  
Dateiname: aufgaben-debugging-prog-c-v1.docx  
Speicherdatum: 18.09.2024  
Autor: Andrew Surber

## Inhaltsverzeichnis

1	Einleitung.....	3
1.1	Sinn und Zweck.....	3
1.2	Beabsichtigtes Vorgehen .....	3
2	Aufgabe 1 .....	4
2.1	Quellcode.....	4
2.2	Hypothese.....	4
2.3	Dokumentation des Problems .....	4
3	Aufgabe 2 .....	5
3.1	Quellcode.....	5
3.2	Hypothese.....	5
3.3	Dokumentation des Problems .....	5
4	Aufgabe 3 .....	6
4.1	Quellcode.....	6
4.2	Hypothese.....	6
4.3	Dokumentation des Problems .....	6
5	Aufgabe 4 .....	7
5.1	Quellcode.....	7
5.2	Hypothese.....	7
5.3	Dokumentation des Problems .....	7
6	Aufgabe 5 .....	8
6.1	Quellcode.....	8
6.2	Hypothese.....	8
6.3	Dokumentation des Problems .....	8
7	Aufgabe 6 .....	9
7.1	Quellcode.....	9
7.2	Hypothese.....	9
7.3	Dokumentation des Problems .....	9
8	Aufgabe 7 .....	10
8.1	Quellcode.....	10
8.2	Hypothese.....	10
8.3	Dokumentation des Problems .....	10
9	Aufgabe 8 .....	11
9.1	Quellcode.....	11
9.2	Hypothese.....	11
9.3	Dokumentation des Problems .....	11
10	Aufgabe 9 .....	12
10.1	Quellcode.....	12
10.2	Hypothese.....	12
10.3	Dokumentation des Problems .....	12
11	Aufgabe 10.....	13
11.1	Aufgabenstellung.....	13
11.2	Hypothese.....	13
11.3	Dokumentation des Problems .....	14

## Änderungsgeschichte

Version	Datum	Autor	Details
1.0	10.09.2020	sura	Initialversion

Tabelle 1 Änderungsgeschichte

## 1 Einleitung

### 1.1 Sinn und Zweck

Das vorliegende Dokument beinhaltet Aufgaben, mit denen das systematische Vorgehen in der Fehlersuche in C, und damit auch allgemein in der Programmierung, geübt werden kann. Insbesondere soll hiermit der Debugger besser kennengelernt werden – in der Theorie, sowie in der Anwendung. Da aber der Debugger selbst in diesem Dokument nicht genauer beschrieben wird, besteht die Absicht, dass dieser im Voraus schon einmal erklärt und vorgeführt wurde.

Die Aufgaben sind ungefähr nach Schwierigkeit geordnet, können aber in beliebiger Reihenfolge gelöst werden. Die Problemstellung sollte anhand des Quellcodes bzw. die davon generierte Ausgabe herausgelesen werden können. Lösungen können eingeholt und bei Bedarf auch im Plenum besprochen werden.

### 1.2 Beabsichtigtes Vorgehen

Für die Erarbeitung der Lösungen ist folgendes Vorgehen verbindlich:

- Schritt 1: Analyse des gegebenen Quellcodes
- Schritt 2: Ausführen des gegebenen Quellcodes in der persönlichen IDE
- Schritt 3: Analyse der generierten Ausgabe in der Konsole
- Schritt 4: Erstellen einer Hypothese: Was ist der Fehler bzw. die Ursache?
- Schritt 5: Durchlaufen des Programms im Debugging-Modus
- Schritt 6: Die gefundene fehlerhafte Stelle dokumentieren (z.B. Screenshot der IDE)
- Schritt 7: Beheben des Problems im Code
- Schritt 8: Verifizieren / Falsifizieren der zuvor erstellten Hypothese

## 2 Aufgabe 1

### 2.1 Quellcode

```
#include <stdio.h>

int main() {
    char selection = '\0';
    scanf(" %c", &selection);

    switch(selection) {
        case 'a':
            printf("The quick brown fox jumps over a lazy dog.");
        case 'b':
            printf("Sphinx of black quartz, judge my vow.");
        case 'c':
            printf("Pack my box with five dozen liquor jugs.");
        default:
            printf("These are called pangrams.");
    }
    return 0;
}
```

### 2.2 Hypothese

Wahrscheinlich stimmt etwas mit dem \0 nicht.

---

---

---

### 2.3 Dokumentation des Problems

Beim programmieren dieses Scripts hat man die breaks vergessen nach jedem case darum sprint es immer direkt zum nächsten.

### 3 Aufgabe 2

#### 3.1 Quellcode

```
#include <stdio.h>

int main() {
    int inMillis = 0;

    printf("Strecke in Millimeter: ");
    scanf("%d", &inMillis);

    int outCentis = inMillis / 10;
    int outMeters = inMillis / 1000;

    printf("%dmm = %dcm = %dm", inMillis, outCentis, outMeters);

    return 0;
}
```

#### 3.2 Hypothese

Es gibt keine Kommastellen an weil es ein Integer ist.

---

---

---

---

#### 3.3 Dokumentation des Problems

Zumachen dass es die Kommastellen anzeigt wie zum Beispiel: 20.00mm = 2.00cm = 0.02m

Dann muss man einen float machen wie zum Beispiel: float inMillis = 0;

Change all %d to %f

Change all int to float

```
int outCentis = inMillis / 10.0;
```

```
int outMeters = inMillis / 1000.0;
```

man muss wenn man ein int durch ein int rechnet .0 hinzufügen damit es ein float ist.

## 4 Aufgabe 3

### 4.1 Quellcode

```
#include <stdio.h>

int main() {
    char thema[] = "Thermonukleare Reaktion";

    printf("Wortumkehrer");
    for (int i = 30; i > 0; i--) {
        printf("Buchstabe %d: %c\n", i, thema[i]);
    }
    return 0;
}
```

### 4.2 Hypothese

!—sollte i ++ sein

---

---

---

### 4.3 Dokumentation des Problems

Bei der hypothese hab ich nicht gesehen dass wortumkehrer steht. Thermonukleare Reaktion hat nur 22 buchstaben also muss man 30 zu 22 wechseln  
Und operator zu >= wechseln

## 5 Aufgabe 4

### 5.1 Quellcode

```
#include <stdio.h>

int main() {
    char policeName[] = "Stadtpolizei ZH";
    char policeAddress[] = "Bahnhofquai 3";
    int policePhone = 117;

    char pizzaName[] = "dieci Pizza Kurier";
    char pizzaAddress[] = "Hohlstrasse 204";
    int pizzaPhone = 0442422070;

    printf("Wichtige Kontaktdaten:\n-----\n");
    printf("%s\n%s\n%d\n\n", policeName, policeAddress, policePhone);
    printf("%s\n%s\n%d\n\n", pizzaName, pizzaAddress, pizzaPhone);

    return 0;
}
```

### 5.2 Hypothese

Die falschen % und dann buchstabe verwendet.

---

---

---

---

### 5.3 Dokumentation des Problems

Das erste null entfernen bei 0442422070 und es dann wieder hier einfügen:  
`printf("%s\n%s\n0%d\n\n", pizzaName, pizzaAddress, pizzaPhone);`

## 6 Aufgabe 5

### 6.1 Quellcode

```
#include <stdio.h>

int main() {
    int numRepetitions = 50;

    while(1) {
        printf("THE END IS NEVER ");
        numRepetitions--;

        if((numRepetitions = 0)) {
            printf("THE END");
            break;
        }
    }
    return 0;
}
```

### 6.2 Hypothese

Es steht nur 1 = zeichen was offensichtlich falsch ist

---

---

---

---

### 6.3 Dokumentation des Problems

Einfach noch ein = hinzufügen dann sieht es so aus und funktioniert:  
if((numRepetitions == 0)) {



## 7 Aufgabe 6

### 7.1 Quellcode

```
#include <stdio.h>

int main() {
    int num = 0;

    printf("Please enter a number between 12 and 42, but NOT 33: ");
    scanf("%d", &num);

    if(num >= 12 || num <= 42 && num != 33) {
        printf("Success! Your number was %d", num);
    } else if (num < 12) {
        printf("Error! %d is below 12", num);
    } else if (num == 33) {
        printf("Really? I told you not to", num);
    } else {
        printf("Error! %d is above 42", num);
    }
    return 0;
}
```

### 7.2 Hypothese

Das ist falsch weil es oder ist am anfang und wenn es 33 ist kommt man gar nicht rein also  
kann man das weg machen und && zwischen grösservergeliche rein tun

---

---

### 7.3 Dokumentation des Problems

Hypothese fast richtige. Dass num != 33 bleibt da aber zwischen dem grösser und kleiner als kommt auch ein && hin.

## 8 Aufgabe 7

### 8.1 Quellcode

```
#include <stdio.h>

int main() {
    int a, b, c = 12;
    int i, j, k = 0;

    printf("a + b + c = a * 3 = b * 3 = c * 3 = 36\n");
    printf("%d = %d = %d = %d\n\n", a + b + c, 3*a, 3*b, 3*c);

    printf("i + j + k = i * 3 = j * 3 = k * 3 = 0\n");
    printf("%d = %d = %d = %d\n\n", i + j + k, 3*i, 3*j, 3*k);

    return 0;
}
```

### 8.2 Hypothese

Nur c und k werden definiert

---

---

---

---

### 8.3 Dokumentation des Problems

Nur c und k werden definiert

## 9 Aufgabe 8

### 9.1 Quellcode

```
#include <stdio.h>

int main() {
    printf("Beholdeth! This shall compileth with nay problems!");
    return 0;
}
```

### 9.2 Hypothese

schreibfehler

---

---

---

### 9.3 Dokumentation des Problems

Die probleme sind nur die schreibfehler

Nicht die richtigen sonderzeichen

## 10 Aufgabe 9

### 10.1 Quellcode

```
#include <stdio.h>

void swapValues(int num1, int num2);

int main() {
    int a = 3, b = 5;
    printf("a = %d and b = %d\n", a, b);
    swapValues(a, b); // swap values of variables a and b
    printf("a = %d and b = %d\n", a, b);

    return 0;
}

void swapValues(int num1, int num2) {
    int temp = num1;
    num1 = num2;
    num2 = temp;
}
```

### 10.2 Hypothese

Es wird werden nicht die richtigen nummern zurück gegeben

---

---

---

---

### 10.3 Dokumentation des Problems

Man musste einen pointer machen damit es funktioniert in der void function mit \* und man muss eine refrence machen oben zum pointer mit & vor der variable

## 11 Aufgabe 10

### 11.1 Aufgabenstellung

```
Meine Aufgabe:
#include <stdio.h>

int main() {
    int array[5] = {1, 2, 3, 4, 5};
    int sum = 0;
    int i;

    for(i = 0; i <= 5; i++) {
        sum += array[i];
    }

    printf("Die Summe der Elemente ist: %d\n", sum);
    return 0;
}
```

### 11.2 Hypothese

---

---

---

---

### 11.3 Dokumentation des Problems