**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: 25.10.2021

Autor: Andrew Surber

Inhaltsverzeichnis

[1 Einleitung 3](#_Toc54252108)

[1.1 Sinn und Zweck 3](#_Toc54252109)

[1.2 Beabsichtigtes Vorgehen 3](#_Toc54252110)

[2 Aufgabe 1 4](#_Toc54252111)

[2.1 Quellcode 4](#_Toc54252112)

[2.2 Hypothese 4](#_Toc54252113)

[2.3 Dokumentation des Problems 4](#_Toc54252114)

[3 Aufgabe 2 5](#_Toc54252115)

[3.1 Quellcode 5](#_Toc54252116)

[3.2 Hypothese 5](#_Toc54252117)

[3.3 Dokumentation des Problems 5](#_Toc54252118)

[4 Aufgabe 3 6](#_Toc54252119)

[4.1 Quellcode 6](#_Toc54252120)

[4.2 Hypothese 6](#_Toc54252121)

[4.3 Dokumentation des Problems 6](#_Toc54252122)

[5 Aufgabe 4 7](#_Toc54252123)

[5.1 Quellcode 7](#_Toc54252124)

[5.2 Hypothese 7](#_Toc54252125)

[5.3 Dokumentation des Problems 7](#_Toc54252126)

[6 Aufgabe 5 8](#_Toc54252127)

[6.1 Quellcode 8](#_Toc54252128)

[6.2 Hypothese 8](#_Toc54252129)

[6.3 Dokumentation des Problems 8](#_Toc54252130)

[7 Aufgabe 6 9](#_Toc54252131)

[7.1 Quellcode 9](#_Toc54252132)

[7.2 Hypothese 9](#_Toc54252133)

[7.3 Dokumentation des Problems 9](#_Toc54252134)

[8 Aufgabe 7 10](#_Toc54252135)

[8.1 Quellcode 10](#_Toc54252136)

[8.2 Hypothese 10](#_Toc54252137)

[8.3 Dokumentation des Problems 10](#_Toc54252138)

[9 Aufgabe 8 11](#_Toc54252139)

[9.1 Quellcode 11](#_Toc54252140)

[9.2 Hypothese 11](#_Toc54252141)

[9.3 Dokumentation des Problems 11](#_Toc54252142)

[10 Aufgabe 9 12](#_Toc54252143)

[10.1 Quellcode 12](#_Toc54252144)

[10.2 Hypothese 12](#_Toc54252145)

[10.3 Dokumentation des Problems 12](#_Toc54252146)

[11 Aufgabe 10 13](#_Toc54252147)

[11.1 Aufgabenstellung 13](#_Toc54252148)

[11.2 Hypothese 13](#_Toc54252149)

[11.3 Dokumentation des Problems 13](#_Toc54252150)

Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Datum** | **Autor** | **Details** |
| 1.0 | 10.09.2020 | sura | Initialversion |

Tabelle 1 Änderungsgeschichte

# Einleitung

## 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.

## Beabsichtigtes Vorgehen

Für die Erarbeitung der Lösungen ist folgendes Vorgehen verbindlich:

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

# Aufgabe 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;  } |

## Hypothese

|  |
| --- |
| Ich dachte, dass das break statement zwischen den cases fehlt |

## Dokumentation des Problems

|  |
| --- |
| #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.")**;** break**;** case 'b':  printf("Sphinx of black quartz, judge my vow.")**;** break**;** case 'c':  printf("Pack my box with five dozen liquor jugs.")**;** break**;** default:  printf("These are called pangrams.")**;** break**;** }  return 0**;** } |

# Aufgabe 2

## 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;  } |

## Hypothese

|  |
| --- |
| Ich dachte,weil die variable vor dem printf geändert wird, printet es den falschen Wert aus. |

## Dokumentation des Problems

|  |
| --- |
| #include <stdio.h>  int main() {  float inMillis = 0**;** printf("Strecke in Millimeter: ")**;** scanf("%f"**,** &inMillis)**;** printf("%fmm ="**,** inMillis)**;** double outCentis = inMillis / 10**;** printf(" %fcm ="**,**outCentis)**;** double outMeters = inMillis / 100**;** printf(" %fm"**,**outMeters)**;** return 0**;** } |

# Aufgabe 3

## 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;  } |

## Hypothese

|  |
| --- |
| Die länge des strings ist verschieden. |

## Dokumentation des Problems

|  |
| --- |
| #include <stdio.h>  int main() {  char thema[] = "Thermonukleare Reaktion"**;** printf("Wortumkehrer")**;** for (int i = 22**;** i >= 0**;** i--) {  printf("Buchstabe %d: %c\n"**,** i**,** thema[i])**;** }  return 0**;** } |

# Aufgabe 4

## 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;  } |

## Hypothese

|  |
| --- |
| Die PizzaPhone Variable wird in okta angezeigt, wegen dem 0 vor der Nummer, des wegen schreibe ich das null vor der variable beim Printf. |

## Dokumentation des Problems

|  |
| --- |
| #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 = 442422070**;** printf("Wichtige Kontaktdaten:\n----------------------\n")**;** printf("%s\n%s\n%d\n\n"**,** policeName**,** policeAddress**,** policePhone)**;** printf("%s\n%s\n0%d\n\n"**,** pizzaName**,** pizzaAddress**,** pizzaPhone)**;** return 0**;** } |

# Aufgabe 5

## 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;  } |

## Hypothese

|  |
| --- |
| Das while(1) ist immer true und beim if statement muss das == sein statt ein =, weil es sonst eine Value zugefügt wird. |

## Dokumentation des Problems

|  |
| --- |
| #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**;** } |

# Aufgabe 6

## 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;  } |

## Hypothese

|  |
| --- |
| Es hat der AND operator gefehlt. |

## Dokumentation des Problems

|  |
| --- |
| #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**;** } |

# Aufgabe 7

## 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;  } |

## Hypothese

|  |
| --- |
| Man kann die variablen nicht so deklarieren |

## Dokumentation des Problems

|  |
| --- |
| #include <stdio.h> int main() {  int a = 12**;** int b = 12**;** int c = 12**;** int i = 0**;** int j = 0**;** int 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**;** } |

# Aufgabe 8

## Quellcode

|  |
| --- |
| #include ˂stdio.h˃  int main() {  printf("Beholdeth! This shall compileth with nay problems!");  return 0;  } |

## Hypothese

|  |
| --- |
| Ich sah, dass das header file falsch ist deswegen habe ich das header file neu geschrieben. Dieses mal hat das funktioniert, weil es keine versteckte charaktere hat oder es die korrekten Zeichen sind. Und das semicolon war auch ein falscher charakter |

## Dokumentation des Problems

|  |
| --- |
| #include <stdio.h>  int main() {  printf("Beholdeth! This shall compileth with nay problems!")**;** return 0**;** } |

# Aufgabe 9

## 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;  } |

## Hypothese

|  |
| --- |
| Die Daten müssen durch pointers abgerufen werden. |

## Dokumentation des Problems

|  |
| --- |
| #include <stdio.h>  void swapValues(int \*num1**,** int \*num2) {  int temp = \*num1**;** \*num1 = \*num2**;** \*num2 = temp**;** }   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**;** |

# Aufgabe 10

## Aufgabenstellung

|  |
| --- |
| Definiere selbstständig ein C Problem, welches mittels Debugging gelöst werden kann und tausche dich mit einer / einem Mitlernende/n aus.  Beachte: Der von dir verfasste Quellcode sollte syntaktisch korrekt sein, semantisch aber Fehler aufweisen. Hierbei wird natürlich vorausgesetzt, dass du noch den Unterschied zwischen Syntax und Semantik weisst.  Zusätzlich wird in dieser Aufgabe erwartet, dass du als Teil der Problemdokumentation die erhaltene Problemstellung auch bewertest. |

## Hypothese

|  |
| --- |
|  |

## Dokumentation des Problems

|  |
| --- |
|  |