

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: 13.09.2024
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	6
4	Aufgabe 3.....	6
4.1	Quellcode.....	6
4.2	Hypothese.....	6
4.3	Dokumentation des Problems	7
5	Aufgabe 4.....	7
5.1	Quellcode.....	7
5.2	Hypothese.....	8
5.3	Dokumentation des Problems	8
6	Aufgabe 5.....	8
6.1	Quellcode.....	8
6.2	Hypothese.....	9
6.3	Dokumentation des Problems	9
7	Aufgabe 6.....	10
7.1	Quellcode.....	10
7.2	Hypothese.....	10
7.3	Dokumentation des Problems	11
8	Aufgabe 7.....	11
8.1	Quellcode.....	11
8.2	Hypothese.....	12
8.3	Dokumentation des Problems	12
9	Aufgabe 8.....	12
9.1	Quellcode.....	12
9.2	Hypothese.....	13
9.3	Dokumentation des Problems	13
10	Aufgabe 9.....	14
10.1	Quellcode.....	14
10.2	Hypothese.....	14
10.3	Dokumentation des Problems	15
11	Aufgabe 10.....	16
11.1	Aufgabenstellung	16
11.2	Hypothese.....	16
11.3	Dokumentation des Problems	16

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>

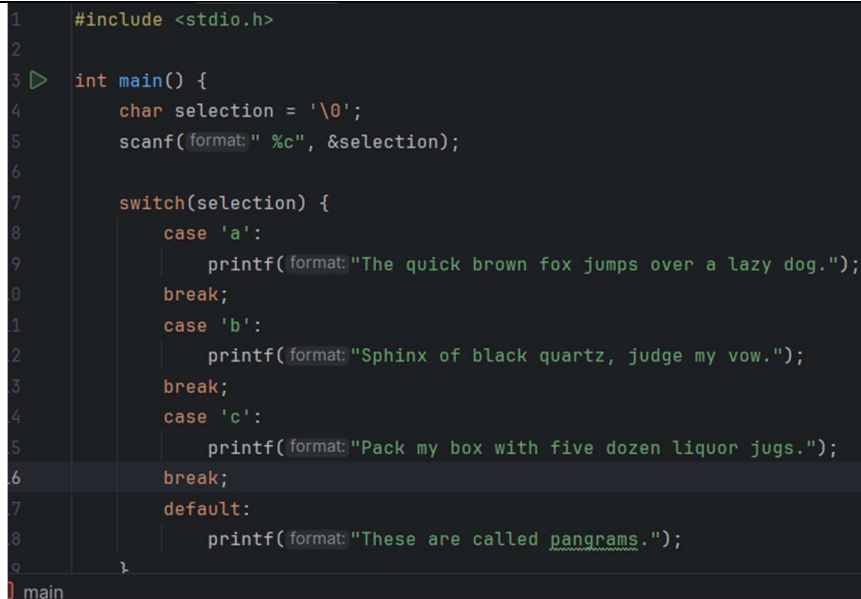
int main() {
    char selection = '\0';
    scanf(" %c", &selection);

    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

Problem war das beim eingeben einer Buchstabe kamen alle sätze vor, obwohl es nicht so sein musste. Das problem sollte irgendwo bei den Sätzen liegen.

2.3 Dokumentation des Problems



```
1  #include <stdio.h>
2
3  int main() {
4      char selection = '\0';
5      scanf(" %c", &selection);
6
7      switch(selection) {
8          case 'a':
9              printf("The quick brown fox jumps over a lazy dog.");
10             break;
11             case 'b':
12                 printf("Sphinx of black quartz, judge my vow.");
13                 break;
14                 case 'c':
15                     printf("Pack my box with five dozen liquor jugs.");
16                     break;
17                     default:
18                         printf("These are called pangrams.");
19             }
20     }
```

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 zeigt nur bei mm etwas deswegen muss man etwas bei den cm und m umändern damit es richtig anzeigt.

3.3 Dokumentation des Problems

```
1  #include <stdio.h>
2
3  int main() {
4      float inMillis = 0;
5
6      printf(format: "Strecke in Millimeter: ");
7      scanf(format: "%f", &inMillis);
8
9      float outCentis = inMillis / 10;
10     float outMeters = inMillis / 1000;
11
12     printf(format: "%fmm = %fcm = %fm", inMillis, outCentis, outMeters);
13
14     return 0;
15 }
16
```

main

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

Dieser Code muss die Thema von unten nach oben buchstabe per Buchstabe schreiben, aber es lässt «T» aus. Ich muss etwas in dem «for (int i = 30; i > 0; i--)» ändern, damit es «T» auch dazu macht.

4.3 Dokumentation des Problems



```
1  #include <stdio.h>
2
3  int main() {
4      char thema[] = "Thermonukleare Reaktion";
5
6      printf(format: "Wortumkehrer");
7      for (int i = 30; i >= 0; i--) {
8          printf(format: "Buchstabe %d: %c\n", i, thema[i]);
9      }
10     return 0;
11 }
12
13
```

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;

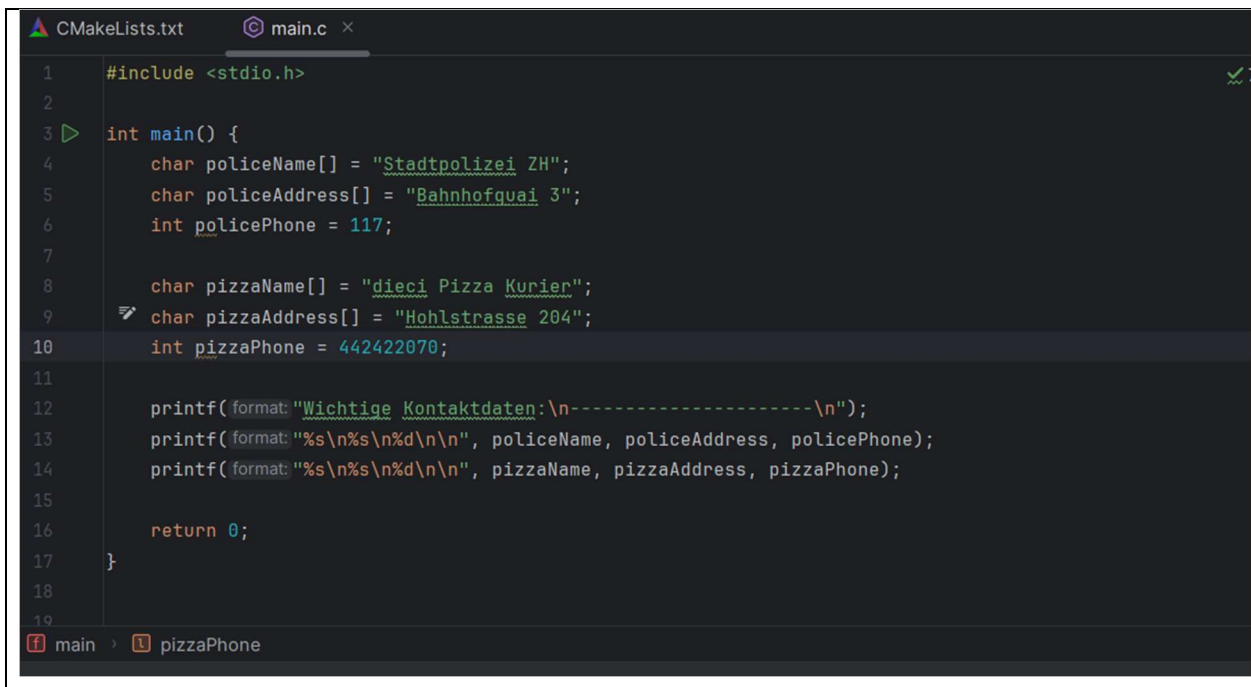
    printf("Wichtige Kontaktdaten:\n-----\n");
    printf("%s\n%s\n%d\n\n", policeName, policeAddress, policePhone);
    printf("%s\n%s\n%d\n\n", pizzaName, pizzaAddress, pizzaPhone);

    return 0;
}
```

5.2 Hypothese

Hier ist das Problem, dass die Telefonnummer von Pizza falsch angezeigt wird. Deswegen muss man etwas bei den Code für Pizza ändern.

5.3 Dokumentation des Problems



```
1  #include <stdio.h>
2
3  int main() {
4      char policeName[] = "Stadtpolizei ZH";
5      char policeAddress[] = "Bahnhofquai 3";
6      int policePhone = 117;
7
8      char pizzaName[] = "dieci Pizza Kurier";
9      char pizzaAddress[] = "Hohlstrasse 204";
10     int pizzaPhone = 442422070;
11
12     printf(format: "Wichtige Kontaktdaten:\n-----\n");
13     printf(format: "%s\n%s\n%d\n", policeName, policeAddress, policePhone);
14     printf(format: "%s\n%s\n%d\n", pizzaName, pizzaAddress, pizzaPhone);
15
16     return 0;
17 }
18
19
```

The screenshot shows a code editor with two tabs: 'CMakeLists.txt' and 'main.c'. The 'main.c' tab is active, displaying the C code above. The code defines variables for police and pizza contact information. The pizza phone number is 442422070. The output shows the police information correctly, but the pizza phone number is displayed as 442422070. The status bar at the bottom shows 'main' and '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

Das Problem liegt daran, dass der code hat keine ende, deswegen muss man etwas bei dem Loop ändern.

6.3 Dokumentation des Problems



```
CMakeLists.txt x main.c x
1  #include <stdio.h>
2
3  int main() {
4      int numRepetitions = 50;
5
6      while(1) {
7          printf(format: "THE END IS NEVER ");
8          numRepetitions--;
9
10         if((numRepetitions == 0)) {
11             printf(format: "THE END");
12             break;
13         }
14     }
15     return 0;
16
17 }
18

main
```

7 Aufgabe 6

7.1 Quellcode

```
#include <stdio.h>

int main() {
    int num = 0;

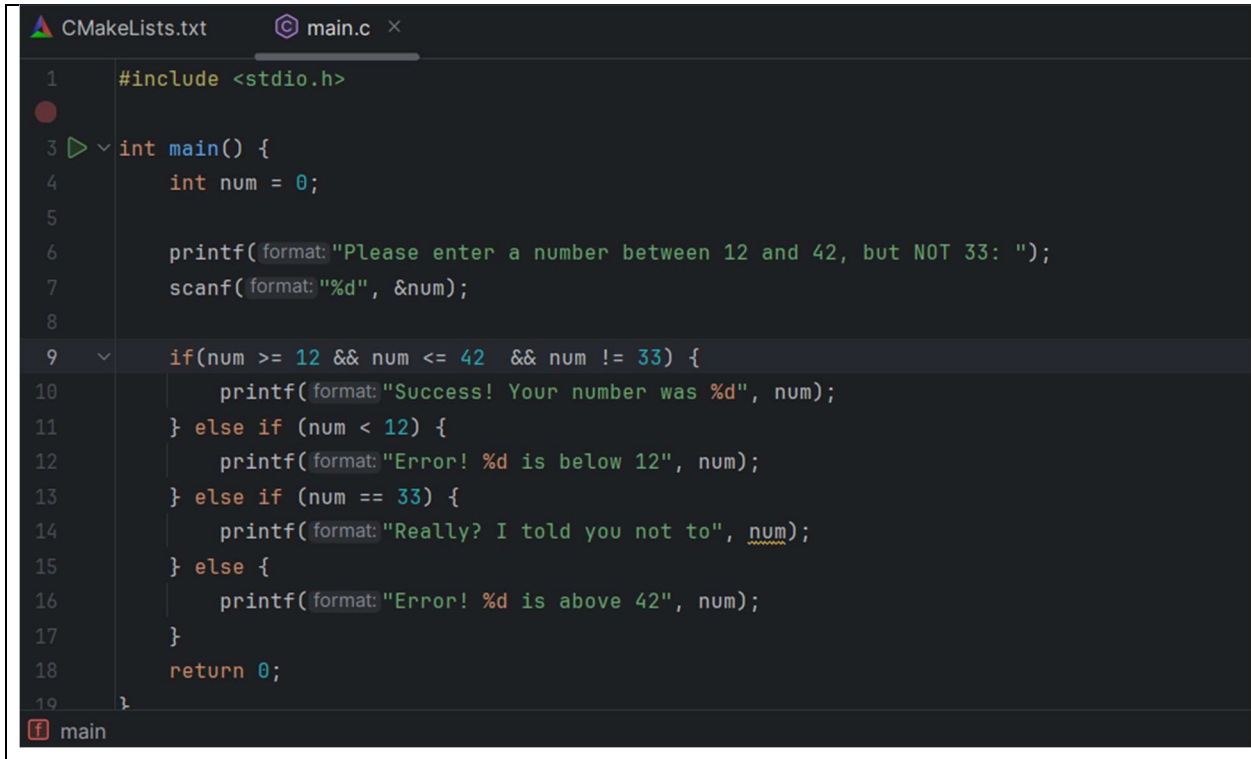
    printf("Please enter a number between 12 and 42, but NOT 33: ");
    scanf("%d", &num);

    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

Es zeigt bei jeder Zahl das man das richtig gemacht hat, obwohl das nicht so sein soll. 33 darf man nicht, unter 12 und über 42 auch nicht. Ich muss etwas in if ändern.

7.3 Dokumentation des Problems



```
1 #include <stdio.h>
3 int main() {
4     int num = 0;
5
6     printf(format: "Please enter a number between 12 and 42, but NOT 33: ");
7     scanf(format: "%d", &num);
8
9     if(num >= 12 && num <= 42 && num != 33) {
10         printf(format: "Success! Your number was %d", num);
11     } else if (num < 12) {
12         printf(format: "Error! %d is below 12", num);
13     } else if (num == 33) {
14         printf(format: "Really? I told you not to", num);
15     } else {
16         printf(format: "Error! %d is above 42", num);
17     }
18     return 0;
19 }
```

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);

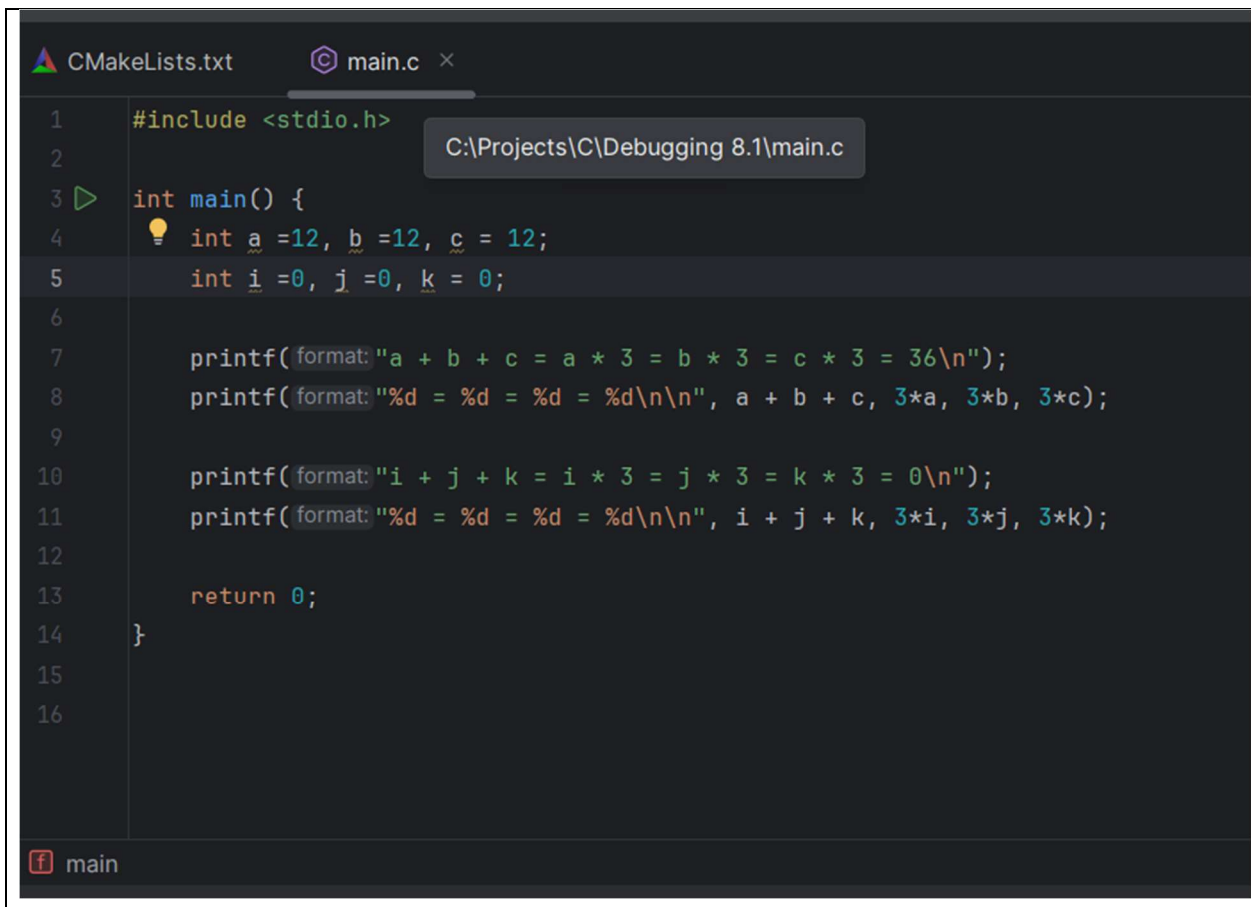
    printf("i + j + k = i * 3 = j * 3 = k * 3 = 0\n");
    printf("%d = %d = %d = %d\n\n", i + j + k, 3*i, 3*j, 3*k);

    return 0;
}
```

8.2 Hypothese

Bei diesem es berechnet nur die buchstabe c und k. Mann muss es so machen damit es auch a,b,i und j berechnet.

8.3 Dokumentation des Problems



```
1  #include <stdio.h>
2
3  int main() {
4      int a =12, b =12, c = 12;
5      int i =0, j =0, k = 0;
6
7      printf(format:"a + b + c = a * 3 = b * 3 = c * 3 = 36\n");
8      printf(format:"%d = %d = %d = %d\n\n", a + b + c, 3*a, 3*b, 3*c);
9
10     printf(format:"i + j + k = i * 3 = j * 3 = k * 3 = 0\n");
11     printf(format:"%d = %d = %d = %d\n\n", i + j + k, 3*i, 3*j, 3*k);
12
13     return 0;
14 }
15
16
```

9 Aufgabe 8

9.1 Quellcode

```
#include <stdio.h>

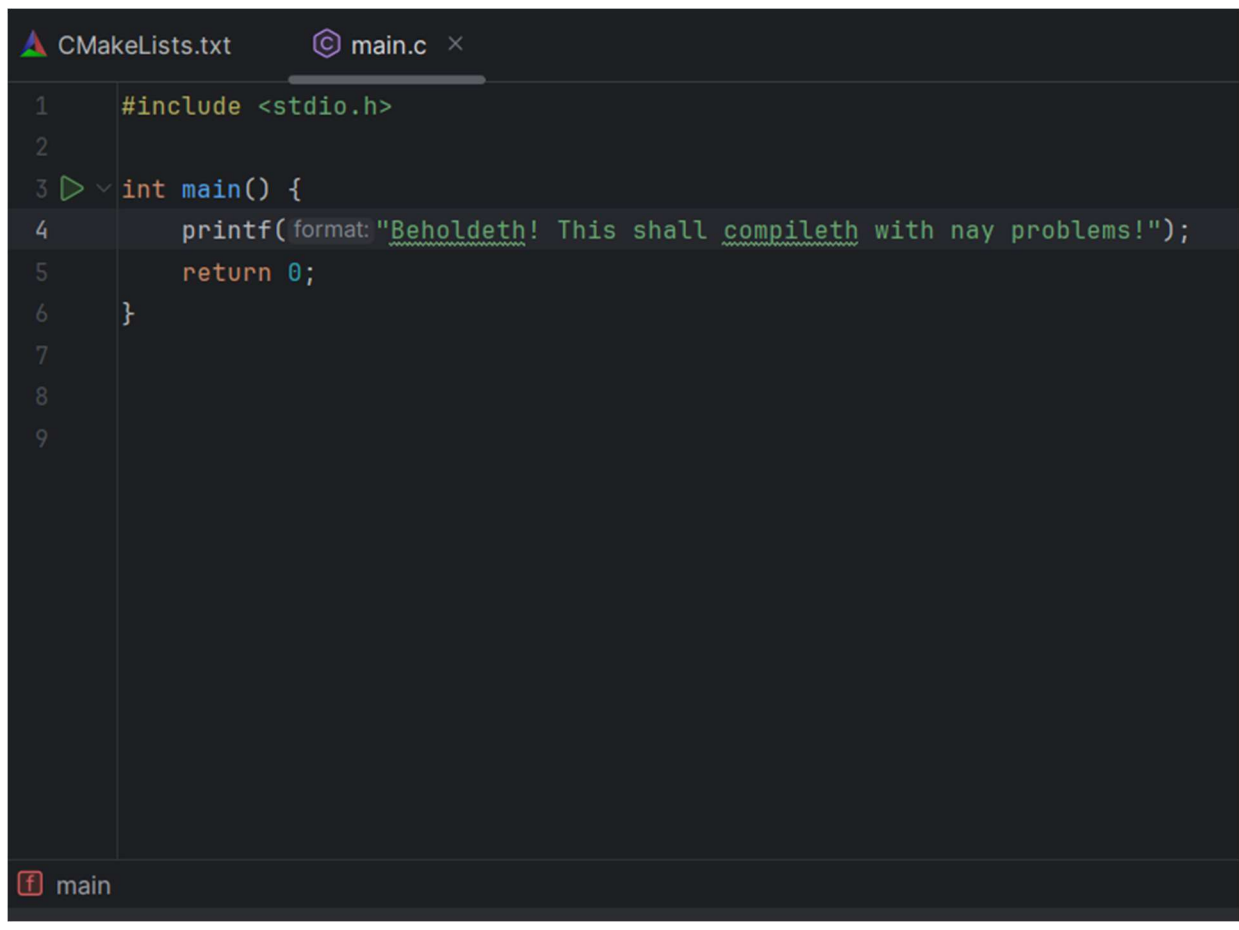
int main() {
    printf("Beholdeth! This shall compileth with nay problems!");
    return 0;
}
```

9.2 Hypothese

Da war alles mit rot am leuchten.

9.3 Dokumentation des Problems

< und ; waren mit einer Variable gemacht, also geklont, faked worden, deswegen ist es nicht gegangen.



The screenshot shows a code editor with two tabs: 'CMakeLists.txt' and 'main.c'. The 'main.c' tab is active, displaying the following code:

```
1  #include <stdio.h>
2
3  int main() {
4  printf(format: "Beholdeth! This shall compileth with nay problems!");
5      return 0;
6  }
7
8
9
```

At the bottom of the editor, there is a status bar showing a red 'f' icon and the text 'main'.

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);

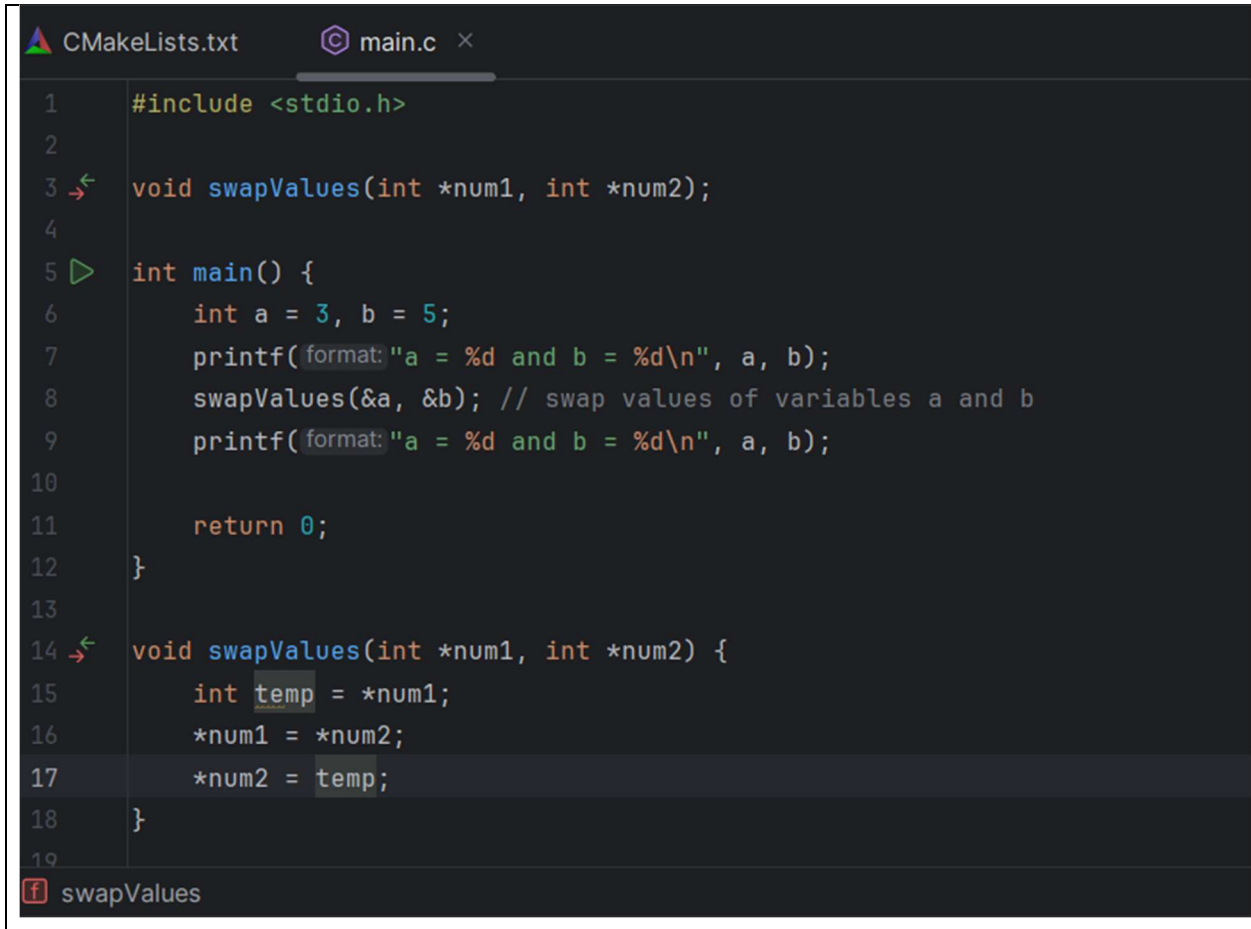
    return 0;
}

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

10.2 Hypothese

Es schreibt beide variante gleich, aber es muss bei zweite verdreht werden.

10.3 Dokumentation des Problems



```
CMakeLists.txt  main.c x
1  #include <stdio.h>
2
3  → void swapValues(int *num1, int *num2);
4
5  ▶ int main() {
6      int a = 3, b = 5;
7      printf(format: "a = %d and b = %d\n", a, b);
8      swapValues(&a, &b); // swap values of variables a and b
9      printf(format: "a = %d and b = %d\n", a, b);
10
11      return 0;
12  }
13
14  → void swapValues(int *num1, int *num2) {
15      int temp = *num1;
16      *num1 = *num2;
17      *num2 = temp;
18  }
19
f swapValues
```

11 Aufgabe 10

11.1 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.

11.2 Hypothese

11.3 Dokumentation des Problems

```
#include <stdio.h>

int main(void) {
    int number = 10;
    printf("Hello, World!\n");
    number = 30;

    printf("Meine Zahl ist : %d %d\n", number, number);

    int age = 28;
    char firstletter = 'n';

    printf("Ich bin %c Jahre alt und mein Name beginnt mit %d\n", age,
firstletter);
    printf("Der Buchstabe %d hat den dez-Wert %c und benoetigt %c Bytes
Speicherplatz\n", firstletter,firstletter, sizeof(firstletter));

    return 0;
}
```