



Leibniz
Universität
Hannover

ASSIGNMENT 05

Programmieren I

21.11.2024

NORWIN BERTRAM WIESECKE

10057811

TIMON WELLHAUSEN

10041137

AUFGABE 1:

Ein Onlineshop bietet Produkte für 1 € pro Einheit an. Für größere Bestellungen wird allerdings ein Mengenrabatt angeboten. Ab einer Bestellmenge von 10 Einheiten kostet jede Einheit nur noch 90 ct, ab 50 Einheiten nur noch 75 ct. Wird nichts oder eine negative Menge bestellt, beträgt der Preis 0 €. Bei einem Bestellwert von unter 20 € wird eine Versandpauschale von 5 € berechnet. Beträgt der Bestellwert 20 € oder mehr, entfällt die Versandpauschale.

Siehe „price.c“.

AUFGABE 2:

Implementieren Sie ein Zahlenratespiel. Der Computer würfelt eine zufällige ganze Zahl im Intervall $[0, 100)$. Der Spieler kann in jeder Runde raten. Wenn die geratene Zahl größer als die Zahl des Computers ist, soll „Too Large!“ ausgegeben werden. Wenn die geratene Zahl kleiner als die Zahl des Computers ist, soll „Too Small!“ ausgegeben werden. Wenn die Zahl der Zahl des Computers entspricht, soll „Match!“ ausgegeben werden.

Siehe „guess_my_number.c“.

AUFGABE 3:

Für die Lesbarkeit von Quelltext ist auch in C die Formatierung sehr wichtig. Quelltext wird häufiger gelesen, als geschrieben. Gegeben sei folgender unformatierter Quelltext:

```
int f ( int i ) { printf ( "called f\n" ) ; if ( i < 0 ) return -i; else return 3*i; }
```

- a) Formatieren Sie diesen Quelltext

Siehe „Aufgabe_3_a.c“.

- b) Erklären Sie das Verhalten der Funktion `f` möglichst kurz und prägnant.

Die Funktion erhält einen Integer `i`. Sie zeigt unabhängig vom Integer den String „called f“ an und erzeugt durch `\n` einen Zeilenumbruch. Ist der Integer kleiner als 0, so wird dieser invertiert und dann ausgegeben. Ist der Integer größer oder gleich 0, so wird dieser mit 3 multipliziert und dann ausgegeben.

- c) Übersetzen Sie den PostFix Code auf der folgenden Seite in schön formatierten C Code, inklusive Purpose Statement und Kommentaren. Benennen Sie die Variablen genauso wie in PostFix, um am Ende genau vergleichen zu können, wie sich die Syntax unterscheidet. Nutzen Sie die obigen Regeln für die Formatierung. Inkludieren Sie die Programmieren 1 Bibliothek und erstellen Sie die Funktion `int main(void)`. Nutzen Sie `test_equal_i` um mindestens 4 sinnvolle Testfälle für die Funktion `number_of_days` zu definieren. Fügen Sie das Programm in der Datei `leap_years.c` Ihrer Abgabe hinzu.

Siehe „leap_years.c“.

AUFGABE 4:

Manchmal spuckt der Compiler seltsame Fehlermeldungen aus. Hier sollen Sie das Programm in primes.c lauffähig machen. Die Funktion `int print_primes_in_intervall(int lower, int upper)` gibt auf der Konsole alle Primzahlen im Intervall `[lower, upper)` aus und gibt als Rückgabewert die Anzahl an gefundenen Primzahlen im Intervall zurück. Wiederholen Sie die nachfolgenden Schritte solange, bis Sie alle Fehler gefunden und behoben haben.

Siehe „primes.c“:

```
primes.c:7:10: fatal error: Bas.eh: No such file or directory
    7 | #include "Bas.eh" //include progl1lib
      |           ^~~~~~
compilation terminated.
make: *** [Makefile:18: primes] Error 1
```

Typo: Bas.eh changed to Base.h

```
primes.c:19:16: error: 'prime' undeclared (first use in this function)
   19 |         while (prime < upper){
      |                ^~~~~
compilation terminated due to -Wfatal-errors.
make: *** [Makefile:18: primes] Error 1
```

Missing { in line 16.

```
primes.c:21:13: error: expected ',' or ';' before '_Bool'
   21 |             bool is_a_prime = true;
      |             ^~~~
compilation terminated due to -Wfatal-errors.
make: *** [Makefile:18: primes] Error 1
```

Missing Semicolon in line 20.

```
primes.c:23:35: error: lvalue required as left operand of assignment
   23 |             if( prime % i = 0 ){
      |                         ^
compilation terminated due to -Wfatal-errors.
make: *** [Makefile:18: primes] Error 1
```

= instead of == in line 23.

```
primes.c:29:13: warning: this 'if' clause does not guard... [-Wmisleading-indentation]
   29 |             if(is_a_prime)
      |             ^~
primes.c:31:17: note: ...this statement, but the latter is misleadingly indented as if it were guarded by the 'if'
   31 |             printf("Die %2d. Primzahl im Intervall: [%d,%d)] ist:
      |             ^~~~~~
%2d\n", count, lower, upper, prime);
primes.c:36:1: warning: control reaches end of non-void function [-Wreturn-type]
   36 | }
      | ^
```

Missing {} in if statement in line 29 and 32(new).

```
primes.c:37:1: warning: control reaches end of non-void function [-Wreturn-  
type]  
    37 | }  
       | ^
```

Function needs to return a value as it is not void. Added „return 0;“.

Bonus: Double semicolon in line 16.