Brett Boggs 1001646678

**1.ai.**

The given output was 32 - -76

**1.aii)**

The next given output was 108

**1.aiii)**

The difference between these two lines of code is the first printf included "%d - %d " vs the second which included "%d ". In our first line, we have **two** different integers that are both being called to print “int1 – int2”(%d - %d). Since -76 is an integer, its ‘-‘ sign stays with it. Our second line that includes "%d" actually will only print **one** integer on a new line but since our integer is an equation it will print the sum.

**1.bi)**

Text

Description automatically generated

**1.bii)**

The reason that these two numbers are printed differently is that while “%d” will print an integer “%f” which is a float, will print a decimal integer.

**1.biii)**

Since we didn’t include “%d” to define our numbers as integers “%f\n” simply prints them as one decimal interger.

**1.biv)**

The difference between these two lines is that the first includes a new line (“\n”) where as the second does not. Their output isn’t affected here.

**1.bv)**

Since the first 12+34 is part of a string and not an integer, it will not be calculated as an integer unlike our second 12 + 34.

**1.bvi)**

See BMB6678Lab1Part1.c

**2.a)**

I get the following error codes

**main.c:19:12: warning: passing argument 1 of ‘printf’ makes pointer from integer without a cast [-Wint-conversion]**

**/usr/include/stdio.h:362:12: note: expected ‘const char \* restrict’ but argument is of type ‘int’**

**main.c:19:5: warning: format not a string literal and no format arguments [-Wformat-security]**

My code finishes with exit code 139

**2.b)** Text

Description automatically generated

**2.c)**

67

**2.d)**

The symbol ‘\*’ causes two integers to multiply

The symbol ‘/’ between two integer numbers causes division between them

**2.e)**

“=” is an assignment operator.

**2.f)**

Text

Description automatically generated

**2.g)**

Displays in the console

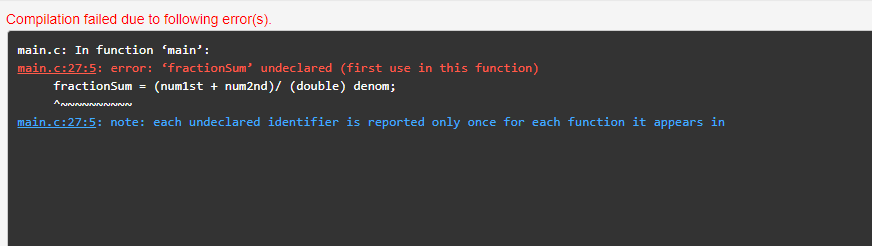
**2.h)**

Age 1, Age 2, Age 3, Age 4

2.i)

Age 5 is different since we use a double which will allow our float to use more decimals.

**3.a)**



**3.b)**

Error on line 27

**3.c)**

“Fractionsum” is undeclared.

**3.d)**

See BMB6678Lab1Part 3.

**3.e)**

Our code did not work since franctionsum included no int behind it. After adding a double the code works fine.

**3.f)**

Text

Description automatically generatedText

Description automatically generatedText

Description automatically generatedText

Description automatically generated