

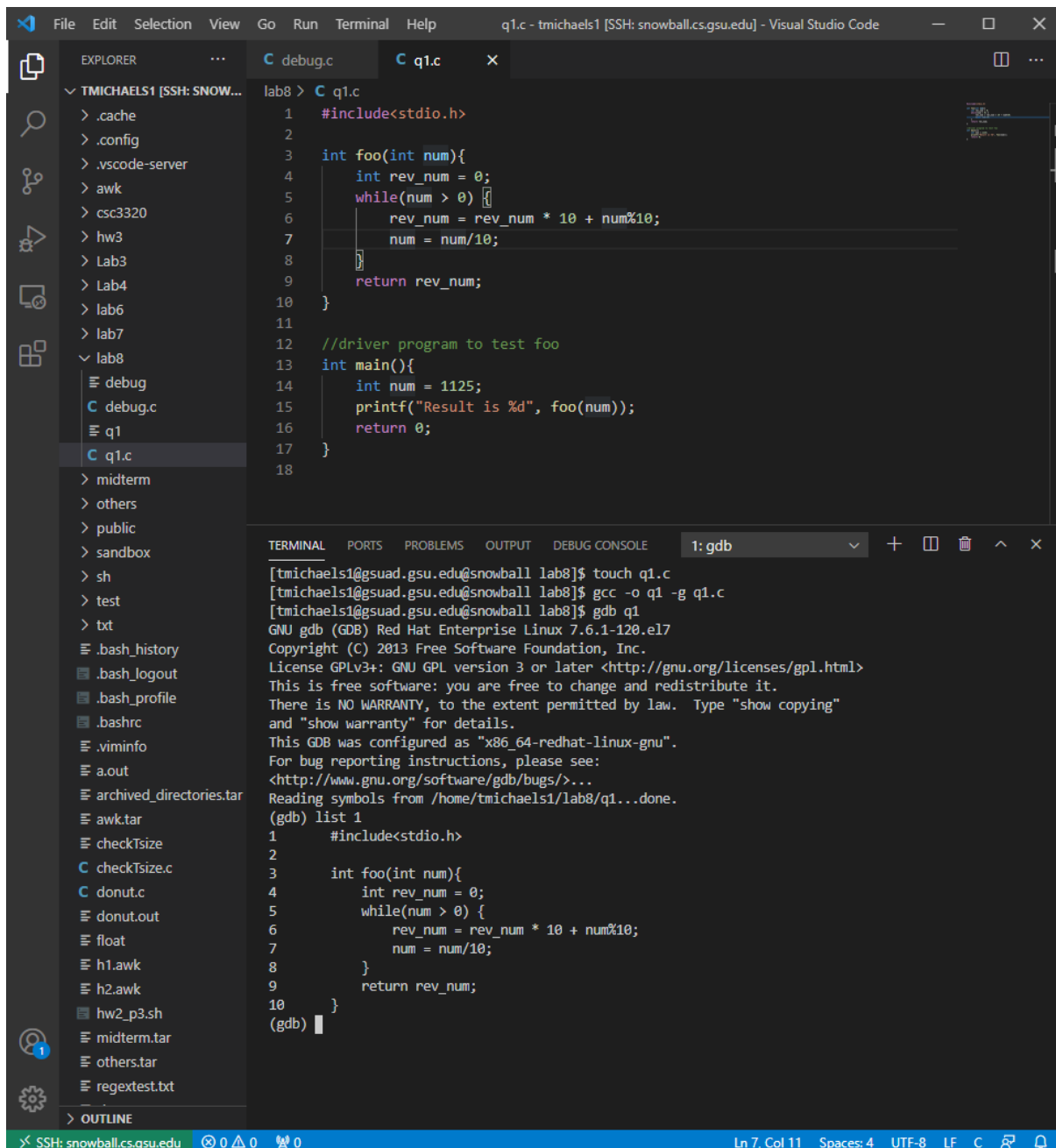
Tracy Michaels

System Level Programming Lab 8

tmichaels1 002430918

Part 1

1-3)



The screenshot shows the Visual Studio Code interface with a C program named `q1.c` open in the editor. The program defines a function `foo` that calculates the reverse of a number by repeatedly reversing its digits. The `main` function tests `foo` with the value 1125 and prints the result.

```
1 #include<stdio.h>
2
3 int foo(int num){
4     int rev_num = 0;
5     while(num > 0){
6         rev_num = rev_num * 10 + num%10;
7         num = num/10;
8     }
9     return rev_num;
10 }
11
12 //driver program to test foo
13 int main(){
14     int num = 1125;
15     printf("Result is %d", foo(num));
16     return 0;
17 }
18
```

The terminal window shows the execution of the program:

```
[tmichaels1@gsuad.gsu.edu@snowball lab8]$ touch q1.c
[tmichaels1@gsuad.gsu.edu@snowball lab8]$ gcc -o q1 -g q1.c
[tmichaels1@gsuad.gsu.edu@snowball lab8]$ gdb q1
GNU gdb (GDB) Red Hat Enterprise Linux 7.6.1-120.el7
Copyright (C) 2013 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later <http://gnu.org/licenses/gpl.html>
This is free software; you are free to change and redistribute it.
There is NO WARRANTY, to the extent permitted by law. Type "show copying"
and "show warranty" for details.
This GDB was configured as "x86_64-redhat-linux-gnu".
For bug reporting instructions, please see:
<http://www.gnu.org/software/gdb/bugs/>...
Reading symbols from /home/tmichaels1/lab8/q1...done.
(gdb) list 1
1     #include<stdio.h>
2
3     int foo(int num){
4         int rev_num = 0;
5         while(num > 0){
6             rev_num = rev_num * 10 + num%10;
7             num = num/10;
8         }
9         return rev_num;
10    }
```

The status bar at the bottom indicates the file is `q1.c` at line 7, column 11, with 4 spaces, UTF-8 encoding, and LF line endings.

4) (gdb) break 5

4-2) (gdb) r

5)

```
(gdb) display rev_num  
1: rev_num = 0  
(gdb) display num  
2: num = 1125
```

6)

	1 st iteration	2 nd iteration	3 rd iteration	4 th iteration
num	112	11	1	0
rev_num	5	52	521	5211

7) (gdb) quit

8) foo function reverses integers passed to it

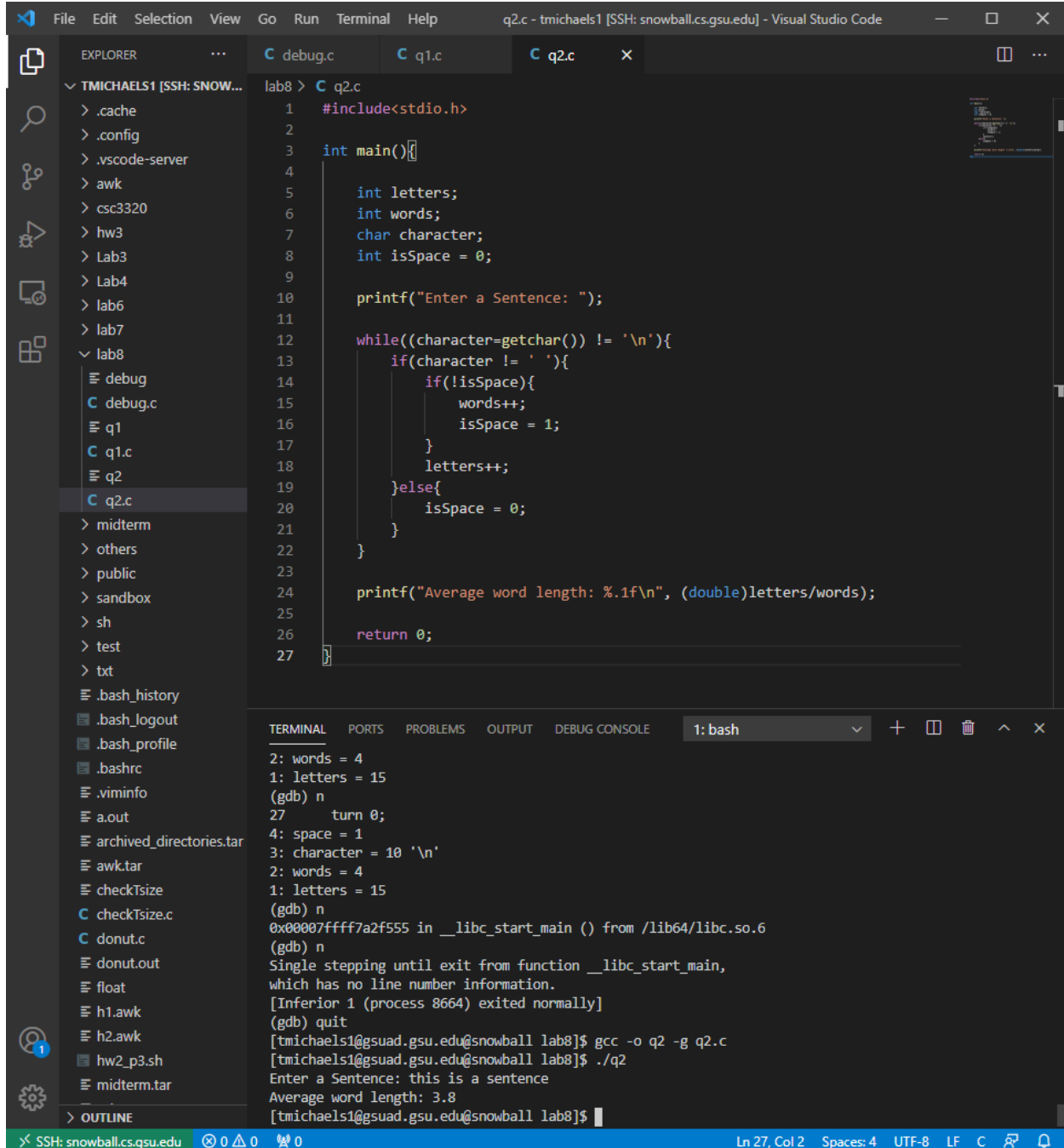
Part 2

Line 11: wrap `\n` in single quotes

Line 13, 15, 19: variable `space` not declared. Added declaration and initialization to 0 at beginning of `main` and changed to `isSpace` for better clarification on what it is

Line 20: `else` statement missing closing bracket

Line 22: cast integers to floating point numbers as to not improperly truncate during division.



The screenshot shows the Visual Studio Code editor with a C program named `q2.c` open. The program calculates the average word length of a sentence. The code is as follows:

```
1 #include<stdio.h>
2
3 int main()
4 {
5     int letters;
6     int words;
7     char character;
8     int isSpace = 0;
9
10    printf("Enter a Sentence: ");
11
12    while((character=getchar()) != '\n'){
13        if(character != ' '){
14            if(!isSpace){
15                words++;
16                isSpace = 1;
17            }
18            letters++;
19        }else{
20            isSpace = 0;
21        }
22    }
23
24    printf("Average word length: %.1f\n", (double)letters/words);
25
26    return 0;
27 }
```

The terminal output shows the program's execution:

```
2: words = 4
1: letters = 15
(gdb) n
27    turn 0;
4: space = 1
3: character = 10 '\n'
2: words = 4
1: letters = 15
(gdb) n
0x00007ffff7a2f555 in __libc_start_main () from /lib64/libc.so.6
(gdb) n
Single stepping until exit from function __libc_start_main,
which has no line number information.
[Inferior 1 (process 8664) exited normally]
(gdb) quit
[tmichaels1@gsuad.gsu.edu@snowball lab8]$ gcc -o q2 -g q2.c
[tmichaels1@gsuad.gsu.edu@snowball lab8]$ ./q2
Enter a Sentence: this is a sentence
Average word length: 3.8
[tmichaels1@gsuad.gsu.edu@snowball lab8]$
```

The status bar at the bottom indicates the file is `q2.c` at line 27, column 2, with 4 spaces, UTF-8 encoding, and LF line endings.

