CSC3320 System Level Programming
Lab Assignment 8 - Post-Lab
Kevin Gallardo

Due at 11:59 pm on Friday, March 12, 2021

Purpose: Learn how to use debugger in gdb to debug a program in Unix.

Part 1:
You are given a C program “q1.c” as below. But since there are no enough comments in the program, it is hard to find out the feature of the function foo. So let us trace the execution of the program and find out what foo does. Please follow the steps below and answer the questions accordingly.

```
#include <stdio.h>

int foo(int num)
{
    int rev_num = 0;
    while (num > 0)
    {
        rev_num = rev_num*10 + num%10;
        num = num/10;
    }
    return rev_num;
}
/* Driver program to test foo */
int main()
{
    int num = 1125;
    printf("Result is %d", foo(num));
    return 0;
}
```

1) Compile “q1.c” with –g option so that we can debug the executable using gdb.

```
$gcc -o q1 -g q1.c
```

2) Lauch gdb for “q1”.

```
$gdb q1
```

3) List the source code of “q1.c” from line 1.

```
(gdb)list 1
```

4) Set a breakpoint at the line of statement “while (num > 0)”.

Question: Write your command.

```
break 6
```

4) Run the program until the first breakpoint.

Question: Write your command.

```
r
```

5) Use display to show the value of rev_num and num at each time when program stops.

```
(gdb)display rev_num
(gdb)display num
```

6) Run the while loop step by step using command n multiple times.

```
(gdb)n
```

Question: check the value of rev_num and num after each iteration and fill in the table below.

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

7) When the program terminates, quit gdb using command q.

```
(gdb)q
```

8) Question: Now can you tell what the function foo does?

The foo function returns the reversed number that gets passed to it as an argument. In the program the number 1125 is passed and foo function returns 5211.

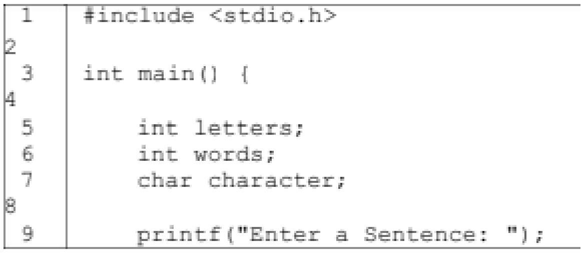
Part 2:

You are given a C program “q2.c” as below. This program is used to calculate the average word length for a sentence (a string in a single line):

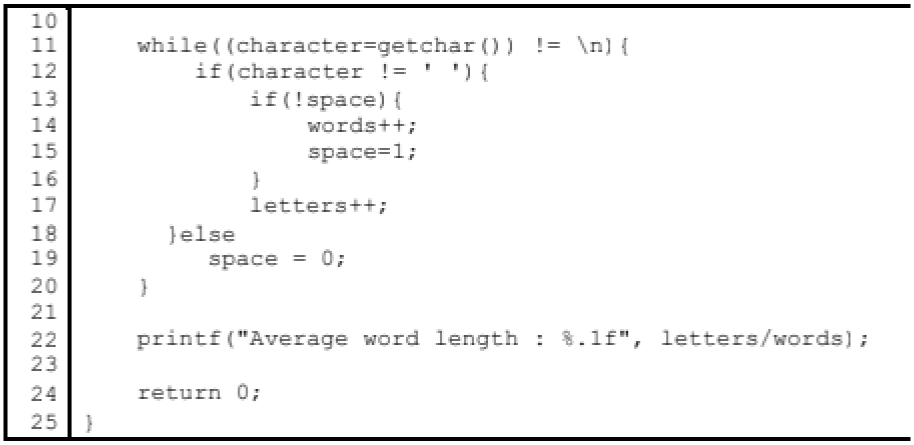
Enter a sentence: It was deja vu all over again.

Average word length: 3.4

For simplicity, the program considers a punctuation mark to be part of the word to which it is attached. And it displays the average word length to one decimal place.



```
1    #include <stdio.h>
2
3    int main() {
4
5        int letters;
6        int words;
7        char character;
8
9        printf("Enter a Sentence: ");
```



```
10
11       while((character=getchar()) != \n){
12           if(character != ' '){
13               if(!space){
14                   words++;
15                   space=1;
16               }
17               letters++;
18         }else
19             space = 0;
20       }
21
22       printf("Average word length : %.1f", letters/words);
23
24       return 0;
25   }
```

However, there are multiple errors in the given C program. Please correct complier errors and use gdb to debug the program and find out the errors.

Question: Please write down the line numbers containing the errors and show how to correct them.

* Line 11: Add single inverted commas around '\n'.

```
while ((character=getchar()) != '\n')
```

* Line 5,6,7: Declare and initialize the variable space. And assign zero to each variable.

```
int letters=0, space=0;

int words=0;

char character=0;
```

* Line 22: Explicitly cast the type in the printf statement to float.

```
printf(“Average word length : %.1f”, (float)letters/words);
```

(Note: you do not need to write down the commands you issued in gdb.)

Submssion:

- Please follow the instructions below step by step, and then write a report by answering the questions and upload the report (named as Lab8_FirstNameLastName.pdf or Lab8_FirstNameLastName.doc) to Google Classroom, under the rubric Lab 8 Out-of-lab Assignment. • Please add the lab assignment NUMBER and your NAME at the top of your file sheet.