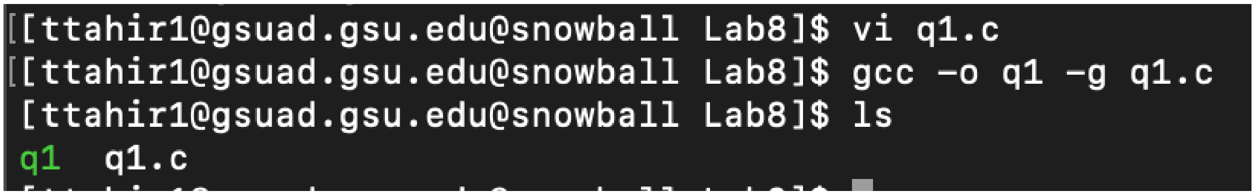
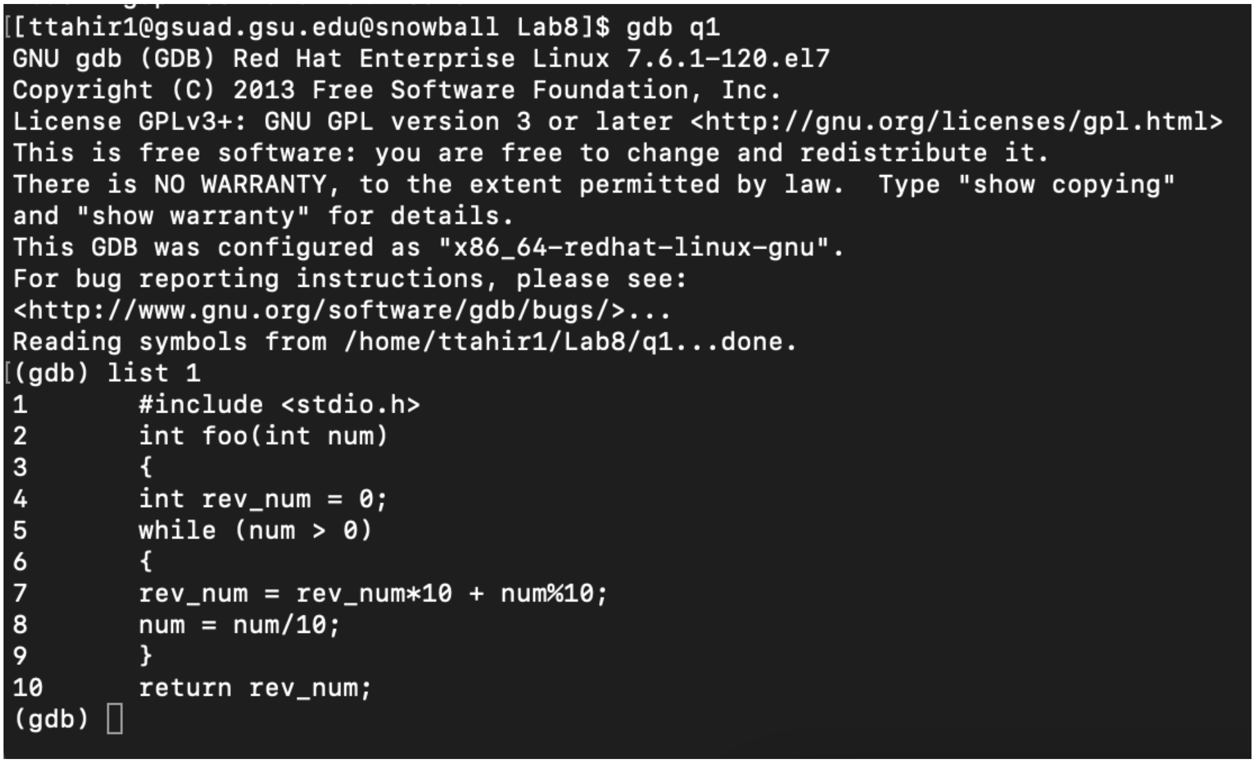
Part 1

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



```
[[ttahir1@gsuad.gsu.edu@snowball Lab8]$ vi q1.c
[[ttahir1@gsuad.gsu.edu@snowball Lab8]$ gcc -o q1 -g q1.c
 [ttahir1@gsuad.gsu.edu@snowball Lab8]$ ls
 q1  q1.c
```

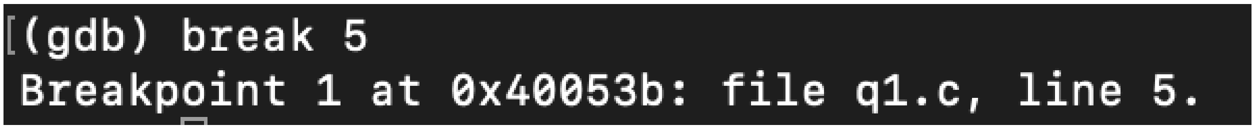
2. Launch gdb for “q1” & List the source code of “q1.c” from line 1.



```
[[ttahir1@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/ttahir1/Lab8/q1...done.
[(gdb) list 1
1       #include <stdio.h>
2       int foo(int num)
3       {
4       int rev_num = 0;
5       while (num > 0)
6       {
7       rev_num = rev_num*10 + num%10;
8       num = num/10;
9       }
10      return rev_num;
(gdb) 
```

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

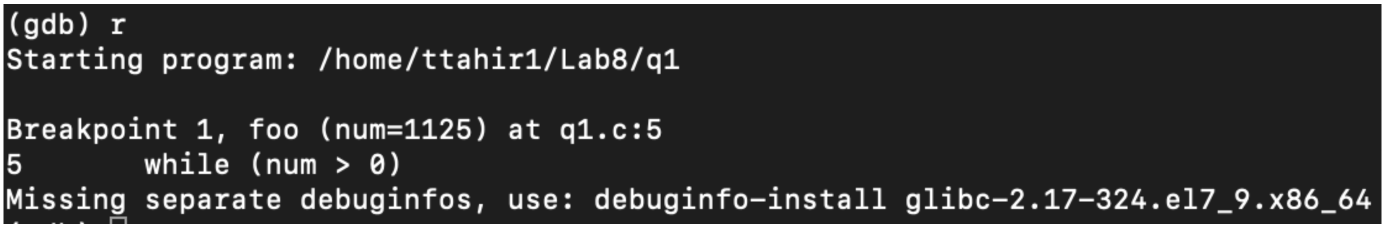
   a. (gdb) break 5



```
[(gdb) break 5
 Breakpoint 1 at 0x40053b: file q1.c, line 5.
```

4. Run the program until the first breakpoint

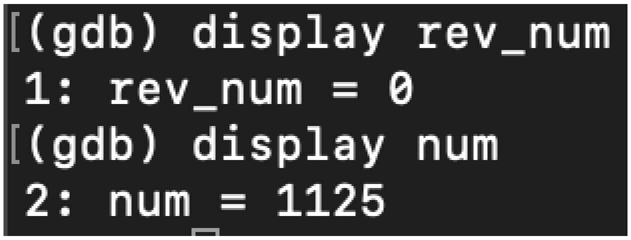
   a. (gdb) r



```
(gdb) r
Starting program: /home/ttahir1/Lab8/q1

Breakpoint 1, foo (num=1125) at q1.c:5
5       while (num > 0)
Missing separate debuginfos, use: debuginfo-install glibc-2.17-324.el7_9.x86_64
```

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



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

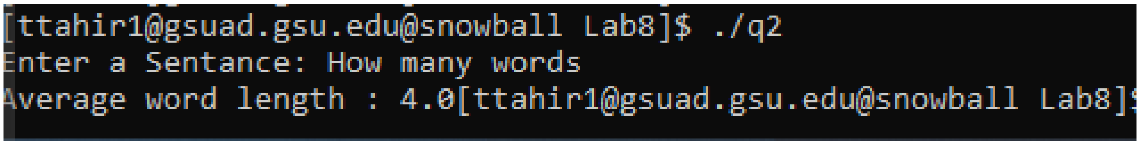
6. Run the while loop step by step using command n multiple times.

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

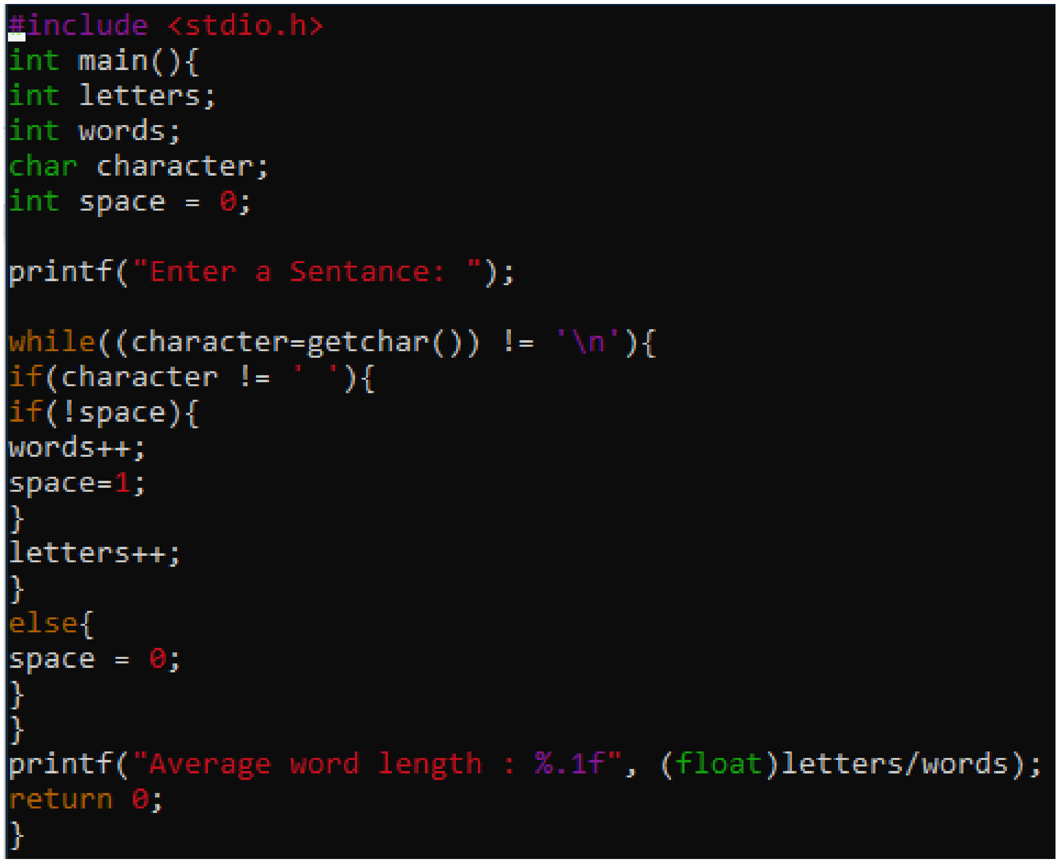
7. Now can you tell what the function foo does?
    a. The function foo takes an integer and finds the rightmost value and puts it into the variable rev_nums, foo does this till the value of num is less than zero. In short the function foo is reversing an integer and returning it,

Part 2

1. Line 8
    a. Error: while((character=getChar()) != \n)
    b. Fixed: while((character=getchar()) != '\n')
2. Line 10
    a. Error: space is not declared
    b. Fixed: int space = 0;
3. Line 12
    a. Error: letters/words are the wrong type
        i. printf("Average word length : %.1f",letters/words);
    b. Fixed: printf("Average word length : %.1f", (float)letters/words);



```
[ttahir1@gsuad.gsu.edu@snowball Lab8]$ ./q2
Enter a Sentance: How many words
Average word length : 4.0[ttahir1@gsuad.gsu.edu@snowball Lab8]$
```



```
#include <stdio.h>
int main(){
int letters;
int words;
char character;
int space = 0;

printf("Enter a Sentance: ");

while((character=getchar()) != '\n'){
if(character != ' '){
if(!space){
words++;
space=1;
}
letters++;
}
else{
space = 0;
}
}
printf("Average word length : %.1f", (float)letters/words);
return 0;
}
```