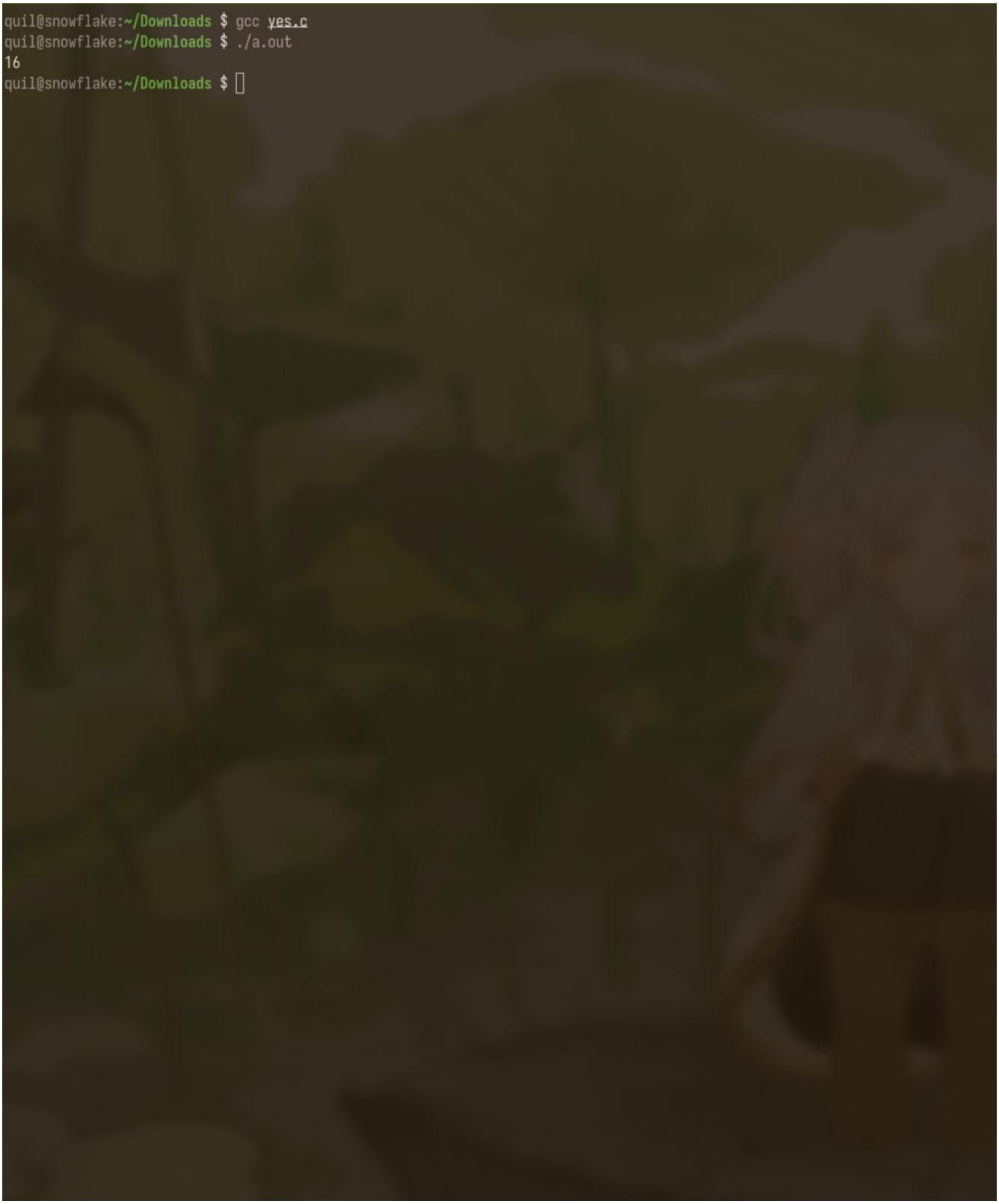
Name: Yousef Awad
Assignment: Portfolio 3
Date: June 23rd, 2025

1.

```c
#include <stdio.h>
#include <string.h>

int main()
{
  char input[] = "This is a sample, examine this sentence for vowels!!!";

  int length = strlen(input);

  int i = 0;
  int count = 0;

  while (i < length)
  {
    if (input[i] == 'a')
    {
      count += 1;
    }
    if (input[i] == 'e')
    {
      count += 1;
    }
    if (input[i] == 'i')
    {
      count += 1;
    }
    if (input[i] == 'o')
    {
      count += 1;
    }
    if (input[i] == 'u')
    {
      count += 1;
    }
    i++;
  }

  printf("%d\n", count);
}
```

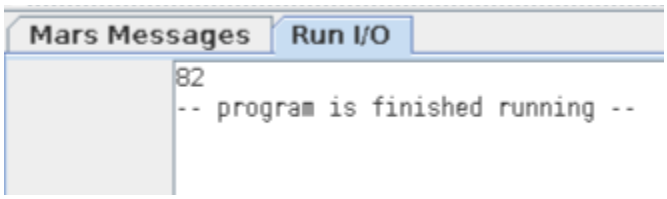
NORMAL yes.c clangd 20% 8:21

"yes.c" 39L, 532B written


quil@snowflake:~/Downloads $ gcc yes.c
quil@snowflake:~/Downloads $ ./a.out
16
quil@snowflake:~/Downloads $

2.

```
# Yousef Awad -- 06/23/2025
#
# strlen.asm
#
# A program that computes
# the length of a string
# similar to the strlen
# found in the C stdlib
#
# Registers Used:
# $t0 - used to hold the loop counter
# $a0 - used to hold the address of the string
# $v0 - syscall parameter and return value

.data
string: .asciiz "This is a sample, find the length of this interesting and null terminating string!"
.text


main:


la $a0, string # string* = string
li $t0, 0 # counter = 0

loop:
lb $t1, 0($a0) # loading next character into t1
beqz $t1, exit # if null, exit

addi $t0, $t0, 1 # counter += 1
addi $a0, $a0, 1 # move string pointer up one
j loop



exit:
# printing amount of characters
li $v0, 1
move $a0, $t0
syscall

# Ending program succesfully
li $v0, 10
syscall

```



3.

strlen.asm | StrtoLowerCase.asm

```
1  # Yousef Awad -- 06/23/2025
2  #
3  # StrtoLowerCase.asm
4  #
5  # A program that shifts
6  # a given string to lower
7  # case characters and returns
8  # it to the user via the console.
9  #
10 # Registers Used:
11 # $t0 - holds address of input string
12 # $t1 - holds current character
13 # $v0 - syscall parameter and return value
14 # $a0 - holds what to print
15
16 .data
17 string:  .asciiz "computer OrGANIzatioN"
18 newline: .asciiz "\n"
19 .text
20
21
22 main:
23
24 la $t0, string # string* = string
25
26 loop:
27 lb $t1, 0($t0) # loading next character into t1
28 beqz $t1, exit # if null, exit
29
30 # if t1 >= A
31 bge $t1, 65, testCase2
32 j nextChar
33
34
35 testCase2:
36 # if t1 > a, nextChar
37 bge $t1, 97, nextChar
38 # t1 is therefore less than a
39 addi $t1, $t1, 32 # making it lowercase
40
41
42 nextChar:
43 sb $t1, 0($t0)
44 addi $t0, $t0, 1 # move string pointer up one
45 j loop
46
47
48
49 exit:
50 # printing amount of characters
51 li $v0, 4
52 la $a0, string
53 syscall
54
55 # printing newline
56 li $v0, 4
57 la $a0, newline
58 syscall
59
60 # Ending program succesfully
61 li $v0, 10
62 syscall
63
```

Line: 14 Column: 28 ☑ Show Line Numbers

Mars Messages | Run I/O

```
computer organization

-- program is finished running --
```

Clear