



Course Name: Electronic Devices Lab  
Course Number and Section: 14:332:333:2A

Experiment: Lab Experiment 1: Introduction to C

Lab Instructor: Jalal Abdulbaqi

Date Performed: 9/26/18

Date Submitted: 10/10/18

Submitted by: Vedanta Dhobley 171002654

### Exercise 1:

1.

We changed the  $V_0$  to '3' in order to print "RU RU RU"

We changed the  $V_1$  to '0' in order to get the case 0 to print the variable  $V_1$  in the switch statement to print out "Werblin Rec Center."

We changed the  $V_3$  to 3 in order for the operator to execute the program and receive a 'true' statement for the expression and prints 'Go.'

We changed  $V_2$  to a number that is NOT 0 in order for the Boolean  $\text{if}(V_2)$  to be true and print the proper information.

2.

The minimum number of distinct values needed for the preprocessor macros is 1. To define a single macro, we need a minimum of 1 value that replaces the constant variable in the executable program.

3.

The -o tag is used to define the executable file name that the gcc makes so that the names of the code is not named "a.out"

### Exercise 2:

1.

We use the command `break[file:] line#` to set a breakpoint at a certain line and use the run command to execute programs.

2.

a) Type 'run' in GDB with the arguments listed as ["and"]. For example. `Run[0, "EXECUTE"]`

b) Write code out as: `break...if(*code*)`

c) 's' goes into function calls and 'n' goes over function calls.

d) If it is a function call, the entire function will be executed in one run (using the answer to c).) Using 's' debugs this problem,

e) 'c' is used for continuous execution of a program.

f) Use the `'print[/f] [expr]'` to see the value of a variable/expression.

g) `display_variable_name` prints out the values of given variables after each step's execution.

h) `display` prints a list of all the variables along with their function values.

i) the command 'QUIT' exits out of the GDB.

### Exercise 3:

1. The while loop (`a != NULL`) was changed to (`a != NULL && b != NULL`). If the LL associated with 'b' is shorter than that of 'a,' it shows a null pointer exception. Otherwise the program will try to access `NULL → val` which does not exist in this program.

### Exercise 4:

1.

We open the GDB after properly executing and compiling the file and using `"run > output_int_hello"` to save the output in the text file named "output\_int\_hello." Entering any input in quotes "..." makes it so that the entire output saves into the text file. We then exit out of the GDB with the 'QUIT' command.

#### Exercise 5:

1.

We start by making sure the list is empty and returned back in its original format if it is empty. Then we define two pointers “tortoise” and “hare” and initialize them to follow the “head” of the linked list. Creating two temporary variables ‘t’ and ‘h’ of the NODE type allows them to be used in the loop that goes through the entire list with the two pointers, ‘hare’ and ‘tortoise.’ If the ‘hare’ pointer reaches a NULL pointer, the linked list is not in a circular path. However, if the hare is equal to the second node in the linked list, this proves that the list is running in a cycle.