RUTGERS

UNIVERSITY

Course Name: _________________

Course Number and Section: **14:332:xxx:xx**

Experiment: [Experiment # [2] – Intro to C programming language]

Lab Instructor:

Date Performed:

Date Submitted:

Submitted by: [Roshni Shah 172005723]

Course Name: ________________

Course Number and Section: **14:332:xxx:xx**

! Important: Please include this page in your report if the submission is a paper submission. For electronic submission (email or Sakai) please omit this page.

-------------------------For Lab Instructor Use ONLY-------------------------

GRADE: ______________

COMMENTS:

Electrical and Computer Engineering Department
School of Engineering
Rutgers University, Piscataway, NJ 08854

Exercise 1:

The changes I made for the program was first seeing how many times “RU” needed to be outputted which was 3 so I went to the loop outputting it and changed the max it was suppose to go up to which was the variable V0 and made it 3. Then I changed V1 to 0 for it to go to the case 0 of “Werblin Rec Center”. I also put a break statement after that case so that once it did that case it would exit the switch. Next, I made the V3 value to 3 because we wanted it to output Go instead of Boo and that would only happen if V3 is equal to 3. Then, I changed V2 to 1 because it just had to be any number besides 0. If it was 0 it would go to the else statement.

Exercise 2:

Once in the gdb for specific file hello, to set up a breakpoint at main, type the command

(gdb) b hello.c:main

1. How do you pass command line arguments to a program when using gdb?

   define

2. How do you set a breakpoint which only occurs when a set of conditions is true (e.g. when certain variables are a certain value)?

   break if [condition]

3. How do you execute the next line of C code in the program after stopping at a breakpoint?

   next

4. If the next line of code is a function call, you'll execute the whole function call at once if you use your answer to #3. (If not, consider a different command for #3!) How do you tell GDB that you want to debug the code inside the function instead? (If you changed your answer to #3, then that answer is most likely now applicable here.)

   step

5. How do you resume the program after stopping at a breakpoint?

   continue

6. How can you see the value of a variable (or even an expression like 1+2) in gdb?

   p variable/expression

7. How do you configure gdb so it prints the value of a variable after every step?

   display variable

8.How do you print a list of all variables and their values in the current function?

print

9.How do you exit out of gdb?

“q” or “quit”

Exercise 3:

The condition only works if a reaches before b but in this program b reaches null before a which makes it a fault. The loop is written to continue while a is not null. In addition, a and b go to the next when the loop runs. For example, when a and b point to nodes[0] both reach null at same time. Another time, b starts further down in the nodes so it reaches null first. But, b reaches null before a, and tries to push it next while null caused a system error. This was fixed by editing the while condition to while ---> while(a!=NULL && b!= NULL)

Exercise 4:

To run the CGDB to completion on the executable made by compiling interactive_hello.c without getting stuck, you must use the standard input and output using redirection. To get the input from a file, you can use the code “a.out < inputfile”

Exercise 5:

```
int ll_has_cycle(node *head) {

node* tort = head; // set starting vals

node* hare = head; //set starting vals

hare = hare->next; //hare move one forward

if (hare==NULL){ //check to make sure hare is not in null

        return 0; //if it happens to be null then no cycle

        }

else {

        hare = hare->next; //hare moves one up (two total)
        tort = tort->next; //tortoise moves one up

if (hare==tort){ //check to see if  they are at the same spot
        return 1; //if they are at same spot then there are cycles so return 1

}

else if (hare==head){ //if hare returns to starting location without reaching NULL or landing on
tort’s space then no cycle happens

        return 0;

}

}

}
// check for code in the zip file in the github
```