



Course Name: Computer Architecture and Assembly language lab

Fall 2018

Course Number and Section: 14:332:333:04

Experiment: [Experiment # [2] –Introduction to C Programming Language]

Lab Instructor: Mengmei Yei

Date Performed: 9/28/2018

Date Submitted: 10/12/2018

Submitted by: [Smeet Kathiria, RUID# 173000011]

Course Name: Computer Architecture and Assembly language lab

Course Number and Section: 14:332:333:04

! 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:

--

. Exercise 1:

Q1. Explain the changes you made?

A- The macros value initially given where 0,0,-1,0 for V0,V1,V2,V3 were changed to 3,3,1,3. Setting macro V0 = 3 will make the for loop run three times and give us a desired output of RU RU RU. Changing the value of V1 to 3 will make the switch statement go to case 3 and will display the output for it. Switch statement will terminate due to break command and the desired output of "Werblin Rec Center is obtained". Setting V3 to 3 will make the comparison argument in the ternary operator true and so the result of a true comparison will be displayed. The changes to the macros value is listed below:

```
#define V0 3
#define V1 3
#define V2 1
#define V3 3
```

2. Explain the minimum number of distinct values needed for the preprocessor macros?

A- The minimum number of distinct values for macros are 1 and 3 because using it will give us the desired output. There are two distinct values needed for preprocessor macros in order to obtain the given output.

```
#define V0 3
#define V1 3
#define V2 1
#define V3 3
```

3. What does the -o flag do with gcc?

A- The -o flag tells the gcc compiler to run the output of the code. For the given assignment we run the code using the line gcc -o simple simple.c

Exercise 2: 1. Explain how do you set the breakpoint at main, and how you run up to that breakpoint.

A- In order to set a breakpoint at main during debugging. We have to compile the file first using the "-g" flag and after that using GNU debugger. I used (b main) command to set a breakpoint at main. Entering this in terminal break the code at main and check if its working or not. After that, In order to check I also set breakpoint at printf and other lines and it worked correctly.

2) Additional gdb commands:

A- Some useful gdb commands are:

b: function, it help us to set breakpoint anywhere in program

run: This command starts our program

bt : backtrace will display program stack

p: expr display the value of an expression

c: continue running your program

gdb program [core]: This command debug program for this purpose which we use is (using core dump core)

n: next line, stepping over function calls

s: next line, stepping into function calls

Exercise 3:

- 1) There was a logical error in the first function. In order to fix it we Because in this code we are checking the variable a and b if they hit the wrong address which is NULL space or not. As our actual code, was only checking the values for A, not for B. To fix this code, we only added a small part inside of while loop which compared (A && B).

Fixed Function:

```
int ll_equal(const node* a, const node* b) {
    while (a != NULL && b != NULL) {
        if (a->val != b->val)
            return 0;
        a = a->next;
        b = b->next;
    }
    /* lists are equal if a and b are both null */
    return a == b;
}
```

Exercise 4:

- 1) I created an input file with the name of Smeet.txt. After making that I downloaded the file from sakai. After that, I used the input command “a.out<input file” to get the desired output.

The Method used is shown below:

```
gcc -g -o int_hello interactive_hello.c
```

./int_hello < Smeet.txt (Smeet.txt) is my input file which I am reading by using command.

```
Smeets-MBP:lab2 smeetkathiria$ ./int_hello < Smeet.txt
```

```
What's your name?
```

```
Hey, Smeet I just really wanted to say hello to you.
```

```
I hope you have a wonderful day.Smeets-MBP:lab2 smeetkathiria$
```

Exercise 5)

- 1) I checked the two pointers, tortoise and hare, if they were NULL. After that I checked if the tortoise and hare pointer were the same. If they are the same, then we would return a true value for both. This loop would keep checking until the tortoise and hare were not equal anymore, which would end the loop.

