

Lab 7

Out: April 6, 2008

Make sure to check on the screen in the front for your name. You will need an email from a Head TA to get credit for this lab if you are not on the list.

Problem 7.1

Now that we're programming in C, you will be using some new tools in lab and to do your homework. First, open a new terminal window to work in. Move into your `course/cs004` directory by typing `cd ~/course/cs004` and make a new directory for this lab by typing `mkdir lab07`. Move into the new directory (type `cd lab07`). You are now ready to begin writing your first C program. Open a new file called `hello.c` by typing:

```
gedit hello.c &
```

This command will open `gedit`, a text editor you can use to edit your code. Type the following code into the `gedit` window:

```
#include <stdio.h>

int main()
{
    printf("Hello, World!\n");
    return 0;
}
```

This code should look similar to code you have seen in lecture. Type the following command in the terminal to compile your code:

```
gcc -Wall -o hello hello.c
```

This command compiles your code and creates an executable file. To run your code, type `./hello` in a shell, and watch what happens.¹ Notice that

¹You do not have to name your executable file the same name as the name of your C file, though this convention often makes the most sense. The string following “-o” is the name of the executable

`./` precedes the name of the program. `./` tells the operating system to look for your program in the current directory; you will always need to preface the name of programs you write with `./` when you want to run them.

For the remaining problems in the lab, we have provided you with stencil code. To get it, type this command into your terminal:

```
cp /course/cs004/pub/lab07/* ./
```

If you run the `ls` command in your `lab07` directory you should now see the files `io.c` and `calc.c` in addition to the files you created for this problem.

Problem 7.2

The file `io.c` currently contains only an include directive and an empty `main()` function. Open the file with `gedit` and modify it to create a program that will prompt a user to enter his/her age and height, store the data in variables of type `int`, and then print out the user's age and height in a sentence.

A sample run of the program might look like this:

```
How old are you?  
> 19  
How tall are you (Round to the nearest foot.)?  
> 6  
You are 19 years old and 6 feet tall.
```

Since you are using variables of type `int`, you will only be able to take height in whole numbers. Remember to properly comment your program.

To compile your program, type `gcc -Wall -o io io.c` in a shell. (When compiling, make sure you are compiling the code when you are in the same directory as your code. You can see what your current directory is by looking at the path starting from the very left of your shell. In this case, your current directory should be `/u/<your login>/course/cs004/lab07.`) To run it, type `./io` and press enter.

Hint: Review the lectures from last week and look at the sections dealing with variable declaration and `printf()` and `scanf()`. Since your data should be of type `int`, you will only need to use `%d` for the I/O calls.

Problem 7.3

In this problem, you will modify the `calc.c` file to create a program that simulates a simple calculator. The program will prompt the user to enter 2 numbers, one at a time. It will then ask the user to enter a number corresponding to an arithmetic operation, with a menu of possible inputs. Make sure you are able to accommodate non-integral decimal operands and invalid user input. Here are 2 sample runs of the program:

```
> ./calc
Enter first number: 4.2
Enter second number: 5
```

```
Enter a 1 for +
2 for -
3 for *
4 for /
Entry: 1
```

The answer is 9.2.

```
> ./calc
Enter first number: 1
Enter second number: 3
```

```
Enter a 1 for +
2 for -
3 for *
4 for /
```

```
Entry: 5
```

```
Invalid operator. Exiting.
```

Hint: You could use if/else statements to do the decision making for this program. However, you will have a cleaner program if you use a switch statement.

Implement your program in the `calc.c` file. To compile, use `gcc -Wall -o calc calc.c`, and to run, type `./calc`. Be sure to fill in the header and to properly comment your program.