CISC105 Spring 2007 Lab05

- Review the code examples from class.
- Some programs below are associated with a question. Answer the questions using C comments below your code in the program file.
- The office hours of the TAs and the instructor are on the class website. Visit us!
- Remember that I expect you to practice programming on your own outside of class assignments (Have a programming weekend! Invite all your relatives!).

Programs

For each numbered problem below you will write a small program. Name each program lab05.n.c, where n is the number in the list below. For example, the name of the file for the first will be lab05.1.c

- 1. Write a program with a function that takes two integer parameters, width and height. The function will print a rectangle of asterisks with the dimensions given by the user. Put the function call inside a loop (what kind of loop?) so you can try new dimensions without exiting the program.
- 2. Make a program with an array of fifty integers. Use a loop to set the values of the array to the multiples of three, starting at zero. Use another loop to print the array, but with two differences. Use a conditional inside the loop to insert a newline after every tenth number (this is similar to the kind of problem you solved in the square problem), and use formatting information (H&K 2.6) to line up the numbers in columns as shown:

> a.out									
0	3	6	9	12	15	18	21	24	27
30	33	36	39	42	45	48	51	54	57
60	63	66	69	72	75	78	81	84	87
90	93	96	99	102	105	108	111	114	117
120	123	126	129	132	135	138	141	144	147

3. Using your indent and asterisk functions from previous labs, make a program that has a menu that lets the user **choose** from four kinds of triangle:

```
* up-left
**
***
***
down-left
**
*
*
*
up-right
**
***
```

```
*** down-right
**
```

After the user selects a type, they will be asked for the size. In addition to the indent and asterisk functions, you will have a separate function for each kind of triangle¹. See the menu-driven problem in the previous lab for more information about the menu.

- 4. Write a program that has a function that uses a for loop to calculate n!, the factorial function. The factorial of 5 is 5 * 4 * 3 * 2 * 1.
- 5. The function rand(), with no parameters, returns a pseudo-random integer on successive calls in the same program. The rand function is located in a special library (not stdio). Look up in your text which library you need to use rand(). Then write a simple program that prints five calls to rand() so you can see how it works.

You can use the mod operator to restrict the range of integers that are produced by calls to rand (try it to make the range 0-10; then 0-100). Then you can shift the whole range to left or right using addition (try 10-20 and -50 - 50).

Write a function that takes as parameters a high and low value, and then produces a random integer in that range (inclusive). Put a call to the function in main, add scanf and a loop so the user can generate as many random numbers as he wishes, each in a different range. Add a prototype if you haven't already.

Now make your output more random: add the statement srand (time (NULL)) to the beginning of your main. Can you see that this is just a function call inside a function call? What do you think it does? Explain in your comments (the answer is hidden in your text).

- 6. Make an array of 10 integers and use rand() to put pseudo-random numbers, ranging from 100 to 500, into the array. Then traverse the array with a loop and find the max element. Print the array (on one line) and the max found with a nice message.
- 7. Here is a formula for computing distance traveled by an object.

 $distance = Velocity_{initial} * time + 1/2 * acceleration * time^{2}$

If we drop an object, it has initial velocity of zero, so the first term goes away. Without the first term in the equation, rewrite the equation to isolate the time variable. You can get help on the rewriting if you need it, but be sure you understand what is happening. Write a function to calculate time, given distance and an acceleration of $9.8meters/sec^2$ (i.e. gravity). Use your function to calculate how long it would take your H&K text to fall from the 102nd floor observatory of the Empire State Building.

The sqrt() function from the math library will be useful. To use it, look up the library you need in your text. *Compile with the -lm option* for the compiler, e.g. gcc -lm file.c

8. Consider the program lab05.8.c available in the lab directory. This program is a hypothetical submission from the midterm, containing a number of errors that I saw more than once.

Here are the instructions from the exam:

¹Are there other ways to solve this? If you'd like to try something different, sketch a solution and get my approval during office hours.

Write a complete program, but do not write any comments unless you think I need them to understand your code. Your program will ask a user for an integer and a double and print the sum of the user's input.

Here are *some* of the point deductions I might use when looking at this kind of problem. NOTE: These deductions may not be a good predictor of deductions on other problems.

wrong type var declaration 1-2 calculation uses garbage 1-2 using undeclared variable 1 improper arguments to a fcn 1-2 poor variable names 1 misleading variable names 2 main doesn't return zero 1 missing & 1 missing #include 1 magic numbers 1-2 missing braces 1 no indent 1-5 extra characters in format string 1 poor prompt 1 incorrect prompt 2 poor output 1-2 missing type cast 1 wrong format specifier 1 unnecessary type cast 1 unnecessary testing 1-3 not following directions 0-all using = instead of == 1-3

Print the program and grade it, writing the reason for the deduction and the number of points you are subtracting.

Then fix every problem you found and submit the program. Since you did not write the code, but just spruced it up, in your comments at the top put your name, and then "modified from code by Terry Harvey".

You should have a total of 8 programs named lab05.1.c to lab05.8.c. Make a single script file (see lab00 for the scripting instructions) where you cat, compile, and run each one in its final form (if it didn't compile, don't run it in the script - mark the place in the printed script file with a colored marker so it stands out). After all files have been run in the script, use ls and cd to show your new directories and their files in the script.

On the first page of every printed copy for this course, your name, section, and TA's name must appear.

Submit all program and script files on MyCourses before midnight Thursday of next week, and give the paper version to your TA at the beginning of your Friday lab (or in lecture Friday if you have a Wednesday lab). Note: cat, compile, and run each program in order! Do *not* cat all programs, then compile, etc.