Lab 3

Lab Overview and Objectives:
1. Practice with rand() and if statements.
2. Practice with #define and the operations +, -, *, /, %, |, &, ^

Prelab:
Print out this page and fill in the correct answers. Bring to your Lab3.

Task 1: Name A File lab3.c:
#include <stdio.h>    /**< included to allow for printf() to be used*/
#include <time.h>     /**< included to allow time() to be used*/
#include <stdlib.h>   /**< include to allow rand() and srand() to be used*/

/** @brief main Function
 * @param void This function does not accept any input variables
 * @return This function would return an error code to the OS if needed.
 */
int main() {
    // variables declarations
    int x;     /**< variable to hold our random integer*/
    srand(time(NULL)); /**< seeds our random number generator. Do this just once */
    x = rand(); printf("x = %i\n", x);
    x = rand(); printf("Now x = %i\n", x);
    x = rand(); printf("Now x = %i\n", x);
}

Compile and run the above program. Do you see that rand() returns random ints?
Edit the program so that
1. it chooses a random int that is always less than 50. I think the mod operator % would be very
   useful here. Anything mod 50 is < 50.
2. it gives the user 5 tries to guess the mystery int. Use scanf statements to read in the user’s
   guesses.
3. it tells the user if they are too high, too low, or exactly right. If they don’t get it right in 5 guesses
   the program just ends. The program also ends if they guess correctly. (Just return 0;)
   Use if statements with else clauses to determine low, high, just right. INDENT YOUR CODE
   INSIDE THE IF STATEMENTS
   Here is my output:

   Guess my number. 25
   25 is too low
   Guess my number. 34
   Perfect guess!
Here is another running of my program:

Guess my number. 25
25 is too low
Guess my number. 35
35 is too low
Guess my number. 42
42 is too high
Guess my number. 39
39 is too high
Guess my number. 37
Perfect guess!

Task 2:
On the command line type the following command. You will be asked to enter 2 ints. Then you will be shown the results of some mystery operations. Try to figure out what the 4 mystery operations are. Your choices are: +, -, *, /, %, &, |, ^
Write down your puzzle code and keep running the code by typing runMysteryOp until you can figure out what the 4 mystery operations are. Write down your guesses.

Get another puzzle (there are 5 random puzzles) and write down its puzzle code and your guesses.

/nfs/stak/a2/classes/eecs/spring2011/cs151/public_html/Labs/lab3/getNewPuzzle

puzzle code _______   mysteryOp1 ___   mysteryOp2 ___   mysteryOp3 ___   mysteryOp4 ___
puzzle code _______   mysteryOp1 ___   mysteryOp2 ___   mysteryOp3 ___   mysteryOp4 ___

F.Y.I. The –E option of gcc lets you see what the preprocessor has done to your code. If you are curious type: gcc –E prog.c and see the results of the #include's and the #defines.

Write a short C program that uses 5 #define statements to define the following:
QUARTERS as * 0.25
DIMES as * 0.10
NICKLES as * 0.05
PENNIES as * 0.01
PLUS as +
Here is your starter code. Name this file Lab3Task2.c You add the #defines at the top. Look at your prog.c from the mystery op exercise for examples of #define.

#include <stdio.h>
int main() {
    float x = 3 QUARTERS PLUS 8 DIMES PLUS 4 NICKLES PLUS 11 PENNIES;
    printf("I have $%.2f\n", x);
    return 0;
}
Check your work by compiling with gcc –E Lab3Task2.c
Then check your work again with a normal compile and run your code. Your output should be:
I have $1.86

Submit Lab3.c and Lab3Task2.c on TEACH before next Monday at 11:59pm.

**Study Questions (written answers due at the beginning of Lab4):**
1. What does the rand() function do?

2. Assume we have int variables beaversScore and ducksScore
   Write an if statement that prints out WE WON!
   if beaversScore is larger than ducksScore, but prints out Wait until next year!
   otherwise.

3. Assume we have these #defines
   
   ```
   #define SPEED_LIMIT 55
   #define FINE 200
   ```
   
   Assume we have an int variable named mySpeed that holds how fast I am going.
   Write an if statement that prints You have to pay $200.
   when mySpeed exceeds SPEED_LIMIT.
   You do not need an else clause. You must use SPEED_LIMIT and FINE. You may define other variables if you want. Do not print out anything if mySpeed does not exceed SPEED_LIMIT.

**Lab 3 Summary:**

<table>
<thead>
<tr>
<th>TASK</th>
<th>Completed?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prelab</td>
<td>4pts.</td>
</tr>
<tr>
<td>Number guessing program works</td>
<td>4pts.</td>
</tr>
<tr>
<td>Indentation is good (indent inside main. indent inside each if)</td>
<td>1pts.</td>
</tr>
<tr>
<td>Correct mystery ops</td>
<td>1pts.</td>
</tr>
<tr>
<td>Correct mystery ops</td>
<td>1pts.</td>
</tr>
<tr>
<td>Correct #defines for QUARTERS, DIMES, NICKLES, PENNIES, PLUS</td>
<td>3pts.</td>
</tr>
<tr>
<td>Study Questions</td>
<td>4pts.</td>
</tr>
</tbody>
</table>

**Extended Learning:**
Modify your lab3.c so that the computer asks YOU to think of a random number between 0 – 50. And the computer gets 5 guesses. Write your program so that the computer always guesses ½ way inside the possible interval. Here is my output when my secret number was 40

Think of a positive number that is less than 50.
My first guess is 25.
Enter 1 for too low, 2 for perfect, 3 for too high. 1
My 2nd guess is 37.
Enter 1 for too low, 2 for perfect, 3 for too high. 1
My 3rd guess is 43.
Enter 1 for too low, 2 for perfect, 3 for too high. 3
My 4th guess is 40.
Enter 1 for too low, 2 for perfect, 3 for too high. 2
Yeah!

Demonstrate your program to a TA at the beginning of Lab4.
Submit your code to Lab3Extra by Monday at 11:59pm.