

Lab 2

Each lab will begin with a recap of last lab and a brief demonstration by the TAs for the core concepts examined in this lab. As such, this document will not serve to tell you everything the TAs will do in the demo. It is highly encouraged that you ask questions and take notes. In order to get credit for the lab, you need to be checked off by the end of lab. For non-zero labs, you can earn a maximum of 3 points for lab work completed outside of lab time, but you must finish the lab before the next lab. For extenuating circumstance, contact your lab TAs and Instructor.

(2 pts) Demo: The Size of Things

Follow along with the TAs as they write a C++ program that first prints the maximum and minimum values for the signed and unsigned ints, longs, and shorts using the macros from the `climits` library, <http://www.cplusplus.com/reference/climits>. Notice there is not a minimum unsigned! You should just enter the numerical literal for this information, and make sure you print proper messages for each min and max you are printing.

(4 pts) Calculate the Size of Things

Make sure you understand the following equations before moving on.

Max Amount of Numbers in Unsigned Binary: 2^x

Max Amount of Numbers in Signed Binary: 2^{x-1}

Maximum value in Unsigned Binary: 2^x-1

Maximum value in Signed Binary: $2^{x-1}-1$

What does x represent?

Why do we subtract the 1 in the exponent?

Why do we subtract the 1 in the expression?

Now, write a program to calculate and print the maximum and minimum signed and maximum and minimum unsigned number stored in n bytes, and **store the result in a variable** to print. You do not have to handle the user entering a number of bytes greater than 8.

How are you going to express an exponent? In C++, you need to use a built-in function, `pow(base, exp)`, from the `cmath` library. For example:

```
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    int num = pow(2,3);
    cout << "2^3 is: " << num << endl;
    return 0;
}
```

Now, insert statements to add 1 to the unsigned and signed maximum number stored in the variable after you calculate the answer, and print the result of increasing the variable by one. Do the same for the unsigned and signed minimum number calculated by subtracting one from the variable and printing the value afterward. What happened? Why?

(4 pts) Prepare for Assignment #2

In your assignment, you have variables, the need to use rand, and conditionals.

Create a program called **rand_numbers.cpp**, and type the following program into the file.

```
#include <iostream>
#include <ctime>      /*included to allow time() to be used*/
#include <cstdlib>   /*include to allow rand() and srand() to be used*/

using namespace std;

int main() {
    int x;          /* variable to hold our random integer*/
    srand(time(NULL)); /*seeds random number generator. Do this just once*/

    x = rand();
    cout << "x = " << x << endl;

    x = rand();
    cout << "x = " << x << endl;

}
```

Compile and run the above program 3 times.

- **What is the result of each rand() for the different executions?**

Comment out the srand() function call. Compile and run the above program 3 times.

- **What is the result of each rand() for the different executions?**

Now add an additional srand() function call in between the two rand/cout lines of code, so there are two srand() function calls. Compile and run the above program 3 times.

- **What is the result of each rand() for the different executions?**

Edit your **rand_numbers.cpp** program so that it **chooses a random int that is in the range 0-5**. I think the mod operator % would be very useful here. What is anything mod 5 or 6? How can this help you? Print the random number to the screen.

Now, use this number, 0-5, to select which message to print:

- If the number is 0, print "Bummer, I'm zero!!!"
- Else if the number is odd, 1, 3 or 5, print "I'm an odd number!"
- Else it's an even number, print "I'm an even number!"

Show your completed work to the TAs for credit. You will not get points if you do not get checked off as an individual.