CS262, Lab 11:
Bitwise Operators
Due: Sunday, Nov 19 at 11:59 pm ET

Description:

In this lab you will write a program that will contain two functions, setlsbs() and getlsbs(). These functions will use bitwise operators to embed and extract "hidden" bits in a character array.

Background Preparation:

- Review bitwise operators (Section 2.9 in textbook).

Specifications:

The 1st function:

```
void setlsbs(unsigned char *p, unsigned char b0)
```

will take as parameters, an array **p** of eight bytes (unsigned char) and a byte **byte0**. It will replace the least significant bits (LSBs) of **p** by the bits of **byte0**. In other words, if the binary representation of **byte0** is $b_7b_6b_5b_4b_3b_2b_1b_0$, you should replace the LSB of **p[0]** by $\mathbf{b_0}$, the LSB of **p[1]** by $\mathbf{b_1}$, ... , and the LSB of **p[7]** by $\mathbf{b_7}$.

The 2nd function:

```
unsigned char getlsbs(unsigned char *p)
```

will take an array **p** of eight bytes (unsigned char) and return a byte **byte0** which is created by combining the LSBs of **p**. That is, your function should combine the least significant bits **bi** of **p[i]** to return a byte $b_7b_6b_5b_4b_3b_2b_1b_0$**.**

Write a program to test your functions as follows:

- Obtain a random number seed from the command line of your program
- Initialize an array p of 8 unsigned char with random numbers from 0 to 255
- Initialize a separate unsigned character byte0 with a random number.
- Print the values in the array p as well as the value for byte0.
  - Print the values in decimal format as well as binary format (use macros defined below)
- Call setlsbs() using p and byte0 as parameters
- After the call to setlsbs() is completed, print the modified values of the array p.
  - Print the values in decimal format as well as binary format (use macros defined below)
- Use the modified array p as a parameter to getlsbs()
- Print the return value of the call to getlsbs(). The returned value should match the original value for byte0
  - Print the value in decimal format as well as binary format (use macros defined below)

Macros:

You may use the following macros to print the binary representation of unsigned character variables:

```
#define BYTETOBINARYPATTERN "%d%d%d%d%d%d%d%d"
#define BYTETOBINARY(byte)  \
  (byte & 0x80 ? 1 : 0), \
  (byte & 0x40 ? 1 : 0), \
  (byte & 0x20 ? 1 : 0), \
  (byte & 0x10 ? 1 : 0), \
  (byte & 0x08 ? 1 : 0), \
  (byte & 0x04 ? 1 : 0), \
  (byte & 0x02 ? 1 : 0), \
  (byte & 0x01 ? 1 : 0)
#define PRINTBIN(x) printf(BYTETOBINARYPATTERN, BYTETOBINARY(x));
```

You can use the macros in a manner similar to the code below:

```
unsigned char num =173;
PRINTBIN(num); printf("\n");
```

Makefile:

Modify the Makefile you used for Lab10 to compile and Lab11

Submission:

You will submit a typescript file similar to the one of previous Labs:

1. Create a typescript file named lab11_typescript_<username>_<labsection>
2. Show that you are logged onto Zeus
3. Show the content of your source code.
4. Compile the code using your Makefile
5. Run your program using 10 different random seeds
6. Remove the executable using your Makefile
7. End the typescript
8. Be sure your directory ONLY contains the *source file*, *script* and Makefile
9. Create a tarfile of your lab11_<username>_<labsection> directory
10. Submit the tarfile to Blackboard
11. Verify that your submitted `tarfile` can be extracted and it's the right `tarfile`.

Congratulations! You have completed your assignment :)