Submission Instructions:

1. For each problem, submit **1)** your program, and **2)** one or two screenshots in **jpg** format showing that your program really works.

2. Name your programs using the pattern NJITID#_ Problem#.c. Name your screenshots using the pattern NJITID#_Problem#_Index.jpg. NJITID# is the eight-digit NJIT ID (Not your UCID, Rutgers students also have NJIT IDs). Problem# is the problem number (e.g., 1, 2, 3, 4, etc). Index is the index number showing the order of the screenshots (e.g., 1 or 2).

3. Submit individual files. **DO NOT SUBMIT A ZIP FILE**.

Objectives

1. To learn how to use bitwise operations and pointers in C programs
2. To understand data binary representation.
3. To learn how to parse command line arguments using getopt().
4. To learn how to handle array of pointers.
5. To learn how to process environment variables.
6. To learn how to handle strings, and pointers pointing to strings.

Problem 1 (30 points)

Write a C program that inverts the bits of a non-negative number provided by the user through argument. Note, the program is to invert the order of the bits, i.e., move the lowest bit to the highest position, 2nd lowest bit to 2nd highest position, etc. It is not to flip the bits (0→1, 1→0).

The program prints out the number, its binary representation, the value after the inversion, and the binary representation after the inversion. Your program can assume that the number is a 32-bit unsigned integer (i.e., a number in $[0, 2^{32} - 1]$).

Refer to the following example for the input and output required by the program, as well as the format.

```
./binary 4891183
4891183:          0000 0000 0100 1010 1010 0010 0010 1111
4098183680:       1111 0100 0100 0101 0101 0010 0000 0000
```

Problem 2 (30 points)

Write a C program that uses `getopt()` to parse its command line. Refer to the example program for `getopt()` in the slides. Assume that you were writing a program named `my_uniq` to report or filter out repeated lines in a file (the same as the *uniq* tool in Linux). The arguments in the command lines are also the same as those for `uniq`:

my_uniq [-c|-d|-u] [-f fields] [-s char] [input_file [output_file]]

Your program is to extract the options, option parameters, and other arguments in the command line.

Note: your program just prints the information on the screen, just as the example program for *getopt*() does in the slides. There is no need to implement the code for reporting or removing repeated lines in the file.

Refer to the manual page of *uniq* (http://man7.org/linux/man-pages/man1/uniq.1p.html) if you need to know what each option means.

Problem 3 (40 points)

Write a C program that sorts the environment variables passed to the program based on environment variable names. In the class, you were shown with a program printing out all the environment variables passed to the program. Each environment variable has a variable name (the part before the '=' sign) and the a variable value (the part after the '=' sign). For example, the following three entries in *envp* (i.e., three environment variables), the names are *USER*, *PWD*, and *HOME*, respectively, and the values are *ubuntu*, */tmp*, and */home/ubuntu*.

```
envp[5] = "USER=ubuntu"
envp[6] = "PWD=/tmp"
envp[7] = "HOME=/home/ubuntu"
```

Your program needs to sort the environment variables. To get the environment variables, your program may use strtok(). The program can directly sort the environment variables by exchanging the pointers saved in *envp*. You may also choose to create another data structure of your choice (e.g., another array of pointers, or linked list). But, no matter which method you choose, your program must print out the environment variables, including their names and values, in ascending order determined by applying *strcmp()* on their names.

For example, the three entries above are sorted by calling *strcmp()* to compare *USER*, *PWD*, and *HOME*. Since the *strcmp()* calls determine that *"HOME" < "PWD"* and *"PWD"<"USER"*, your program should print out

```
HOME=/home/ubuntu
PWD=/tmp
USER=ubuntu
```