Exercise Problems - 8

Data Types, Representations and Ranges of Values

1. Write a program that takes a string of '0's and '1's represnting a positive integer n as input from the user and outputs the decimal value of n (for example, if the input is 1100, the answer should be 12). You may assume that the input string has at most 20 characters.

2. Write a program that takes a string consisting of 8 characters, each of which is '0' or '1', representing a signed binary integer value in 2's complement form and output the equivalent decimal value (for example, if the input is 11000001, the answer should be -63). Do the necessary error checking.

3. Modify the program for Question No.1 so that the result is printed as a hexadecimal number and also as an octal number (hint : use `printf(‘‘%X’’)` and `printf(‘‘%o’’)`).

4. Modify the program in Question No.1 so that it can also handle strings of '0's and '1's and a decimal point (for example, if the input is 1100.101, the answer should be 12.625). You may assume that the string has at most 20 characters.

5. Write a program that takes a string representing a positive real number k in decimal, where $k \leq 1000$ and k has at most 3 digits after the decimal point, as input and outputs its corresponding binary representation. You may truncate the output to a precision of at most 10 bits after the decimal point (for example, if the input is 12.625, the output should be 1100.101).

6. Refer to the definitions made in the library file limits.h for solving this question. Write a program that displays the following values (with respect to your system) and try to understand those values:

   - Number of bytes used to store variables of the following types: `short, int, long int, long long int`.
   - Minimum value that can be represented by a signed `long long int` variable.
   - Maximum value that can be represented by a signed `long long int` variable.
   - Maximum value that can be represented by an `unsigned int` variable.

7. Refer to the definitions made in the library file float.h for solving this question. Write a program that displays the following values (with respect to your system) and try to understand those values:

   - Number of bytes used to represent a `double` variable.
   - Largest positive value a `double` variable can take.
   - Smallest positive value a `double` variable can take.
   - Number of bits in the mantissa while representing the value of a `double` variable.
   - Number of decimal digits of precision for the `double` data type.
   - Maximum positive integer x such that 2^{x-1} can be represented precisely using a `double` variable.
   - Difference between 1 and the smallest positive value greater than 1 that can be represented precisely using a `double` variable.