Week 2 Exercises

1. Write a C++ program that takes in one user input argument from the command line and checks if it is valid hexadecimal number or not, then displays the hexadecimal number to the console.

   - If the incorrect number of user input arguments is detected, the program must exit with an error
   - Hexadecimal values between 0x00 and 0xFF are valid, other values should be rejected
   - The letter "x" of number base can either be upper or lower case
   - The hexadecimal digits can only contain upper case characters.

   Example Run:

   ```
   ./a.exe 0xAB
   > Got valid hexadecimal value: 0xAB
   ```

2. Write a C++ program that takes in up to 5 integer numbers as user input arguments from the command line and stores them into an array of type int. Print out each value to the console.

   A pointer should be used to access each element in the array (to store the user input argument, and output each value in the array to the console).

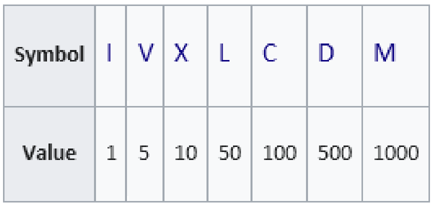
   The program must exist with an error if:

   - Incorrect number of user input arguments
   - At least one user input argument is not valid integer number

   Additional: Edit the code to take in up to 10 floating-point numbers as user input arguments from the command line and stores them into an array of type float. The pointer type will be need to be changed to float

3. Write a C++ program which takes an integer as user input argument from the command line, or ask the user to input the number if it is not given from command line arguments.

   Convert it to a roman number and print out.



Symbol	I	V	X	L	C	D	M
Value	1	5	10	50	100	500	1000

<u>Example Run:</u>

```
./a.exe 12
> Converted Roman number: XII

./a.exe
Input an integer: 12
> Converted Roman number: XII
```

4. Write a C++ program to

   a) Print 200 with and without a positive + sign.

   b) Print 10 and 200 left justified with 5-character width.

   c) Read a hexadecimal value (preceded by 0x) into variable hex, then print out its decimal value.

      Input: 0x10

      Output: 16

   d) Print 100 in hexadecimal form preceded by 0x.

   e) Print 1.234 with 9-character width, padded by zeros.

   f) Print 1.234 with precision is 2 digits of fractional part (after decimal point).

   g) Read a string from the console input, store the string in character array str. Then print out its all characters, separated by a space.