Due: Smartsite Wed., 10/7, 11:55 p.m.
Files to Submit: **changeOfBase.cpp, scientificFloating.cpp, triMatMult.cpp, ReadMe.txt**

- If you are working in a group **ALL** members must submit the assignment
- All programs should compile with no warnings when compiled with the -Wall option
- All prompts for input and all output must match my prompts/output. We use a program to grade your work and tiny differences can cause your work to be marked as a 0.
  - The best way to avoid being deducted points is to copy the prompt/unchanging portion of the outputs into your code. Make sure to get the spaces as well.
- You must also submit a file called ReadMe.txt. Include the names of all partners and any trouble you had on the assignment
- An example ReadMe.txt has been included with this assignment
- The input in the examples has been underlined to help you figure out what is input and what is output
- Submit your work on Smartsite by uploading each file separately. Do not upload any folders or compressed files such as .rar, .tar, .targz, .zip etc.
- If you have any questions please post them to Piazza

1. Write a C++ program called **changeOfBase.cpp** that converts an integer number from one base to another. The program should ask the user for the current base, the number in that current base, and the new base to be converted to. Valid bases are between 2 and 36. Values of digits from 10 – 35 will be represented by characters A – Z. You are guaranteed that all values will be less than or equal to 2^(32) - 1.
   1. Name your executable **changeOfBase.out**
   2. Example 1
      ```
      Please enter the number's base: 10
      Please enter the number: 25
      Please enter the new base: 2
      25 base 10 is 11001 base 2
      ```
   3. Example 2
      ```
      Please enter the number's base: 6
      Please enter the number: 405
      Please enter the new base: 19
      405 base 6 is 7G base 19
      ```
   4. Example 3
      ```
      Please enter the number's base: 24
      Please enter the number: GIG
      Please enter the new base: 10
      GIG base 24 is 9664 base 10
      ```

2. Write a C++ program called **scientificFloating.cpp** that reads in a floating point number and outputs its scientific base 2 format.
   1. Name your executable **scientificFloating.out**
   2. Example 1:
      ```
      Please enter a float: 3.75
      1.111E1
      ```
   3. Example 2:
      ```
      Please enter a float: 140.1
      1.00011000001100110011101E7
      ```
   - You should use bitwise operators to pick out the fields that you need to work with.
   - In order to do the appropriate bitwise operations on the float it must first be cast to an int but (int) f (assuming f is the variable you have stored you float in) will not work as the cast will convert the float representation to the 2's compliment integer representation. The fix is to take the address of the float, cast it as an unsigned int*, and then dereference
     - unsigned int float_int = *((unsigned int*)&f);
3. An upper triangular matrix is a special type of matrix where all the values below the main diagonal are 0. In order to save space we can store this matrix without the zeros. For example

1	2	3
0	4	5
0	0	6

Would be stored as

1	2	3	4	5	6

We would also like to be able to work with these matrices in their compressed format, again to save space. Write a C++ program called, **triMatMult.cpp**,that accepts as arguments two files that contain these compressed upper triangular matrices. The program should multiply the two matrices together and then display the resulting compressed matrix in its compressed form.

- Name the executable **triMatMult.out**
- The names of the files will be given on the **command line**
- All matrices will be square, ie N X N
- All values will be integers
- File format:
  - N (dimension of the matrix)
  - number1
  - number2
  - number3 ...
- For help on matrix multiplication see http://www.purplemath.com/modules/mtrxmult.htm.
- **Restrictions**: You cannot expand the compressed matrices to do the multiplication. If you do this you will receive no credit for this portion of the assignment. Again the whole point is to save space.
- In the examples below the values are shown on 1 line to save space

```
Cat mat1.txt
4
1 2 3 17 4 51 25 6 31 9

cat mat2.txt
4
25 73 -4 -17 -99 81 -88 11 12 10

./triMatMult.out mat1.txt mat2.txt
25 -125 191 13 -396 885 510 66 382 90
```

This is equivalent to doing C = A * B where

A =

1	2	3	17
0	4	51	25
0	0	6	31
0	0	0	9

B =

25	73	-4	-17
0	-99	81	-88
0	0	11	12
0	0	0	10

C =

25	-125	191	13
0	-396	885	510
0	0	66	382
0	0	0	90