Q5. Operator Overloading

For this problem, you need to know how to implement operator overloading of a class.
You must implement the five operators , four member functions and one constructor as follows:

- vector + vector: addition of two vectors.
  - Ex:
    (3, 7) + (-2, 6) = (1, 13)
- vector – vector: subtraction of two vectors.
  - Ex:
    (6, -5) – (3, 3) = (3, -8)
- vector = vector: assign one vector to another vector.
  - Ex:
    v2 = v1
- ostream << vector (print): print the vector in a specific format.
  - Ex:
    If vector = (5, 6), cout << vector will print "(5, 6)".
- ifstream >> vector (read): read the vector from .txt file in a specific format.
  - Ex:
    If the input is "n num1 num2 … numn", which are the dimension and the elements of vector, file >> vector will read the input and store the elements sequentially.

3 6 7 -2	=>	(6, 7, -2)
2 -8 4	=>	(-8, 4)
4 1 2 3 4	=>	(1, 2, 3, 4)
- normalize: normalize the vector.
  - Ex:
    vector = (3, 4) , vector.normalize() will return (0.6,0.8)
- length: calculate the length of vector.
  - Ex:
    vector = (3, 4) , vector.length() will return 5.
- getSize: get the size of vector.
  - Ex:
    vector = (3, 4) , vector. getSize() will return 2.
- AddNumbertoArr(float number): store data into vector.
  - Ex:
    vector. AddNumbertoArr(5) , store 5 into vector.
- Vector(int size): constructor of vector , dynamic allocate an array to store the data.
  - Ex:
    vector v1(6) , create a vector v1, the size of vector is 6.

You must use **operator overloading** to implement.
You must use template to do this lab.
Do not use std::vector.

Input Format

Please implement the file I/O part.

You MUST read the input data from the input.txt.

The first line shows the number of test cases.

Normalization and length operations has two lines:
The first line contains an operator.
The second line is the operand, which is a n-dimension vector.

Other operations have three lines:
The first line contains an operator.
The second line is the operand, which is a n-dimension vector.
The third line is the second operand, which is a n-dimension vector.

P.S. The dimensions of the vectors are the same in each operation and there will not have divide 0 problem when normalize.

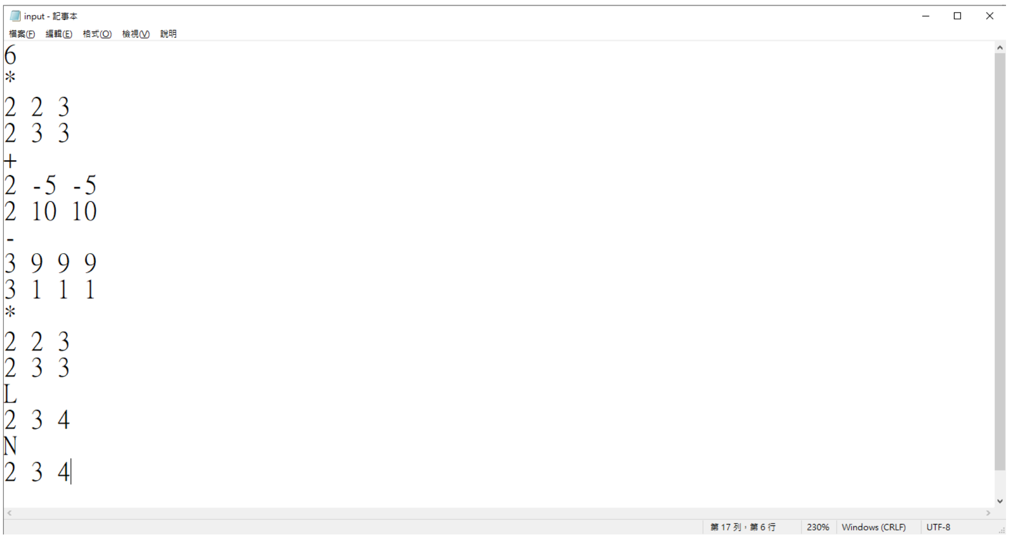
Output Format

You must output the result after doing each calculation.

See more detail from Sample output.

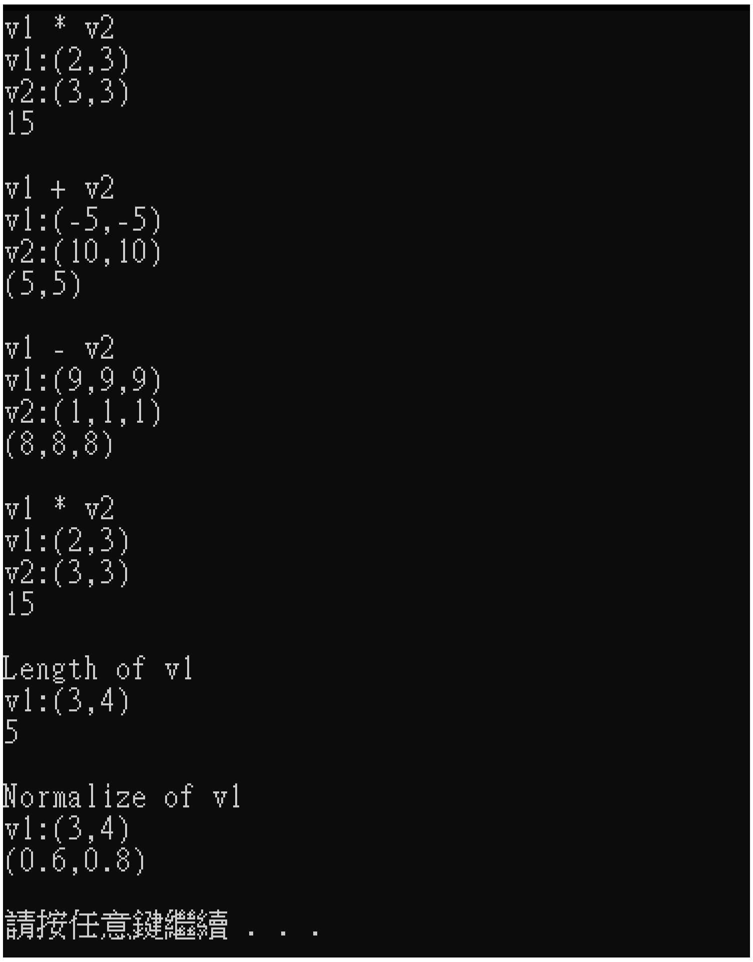
Sample Input & Output.

Input:



```
6
*
2 2 3
2 3 3
+
2 -5 -5
2 10 10
-
3 9 9 9
3 1 1 1
*
2 2 3
2 3 3
L
2 3 4
N
2 3 4
```

Output:



```
v1 * v2
v1:(2,3)
v2:(3,3)
15

v1 + v2
v1:(-5,-5)
v2:(10,10)
(5,5)

v1 - v2
v1:(9,9,9)
v2:(1,1,1)
(8,8,8)

v1 * v2
v1:(2,3)
v2:(3,3)
15

Length of v1
v1:(3,4)
5

Normalize of v1
v1:(3,4)
(0.6,0.8)

請按任意鍵繼續 . . .
```