Q # 1

Vector of vectors!

Let's create a class called *VectorOfVectors* that can stores *n* vectors . Whenever we instantiate an object of this class we pass as parameter the number of vectors this object will store (all one dimensional), an array containing the initial value that we want each vector to be initialized with (optional) and size of each vector (all vectors will have the same size). The following object declarations are valid:

```
int a[3]={10,0, 3};

VectorOfVectors v(3, a, 10);                    //it stores 3 vectors, all of size 10,
                                                //with first vector initialized to 10,
                                                //second 0 and third 3.
VectorOfVectors v(0,10);                        //it stores only 1 vector of size //10
                                                //initialized with 0.
```

Since we know that C++ stores multi-dimensional arrays in row major order, let's implement our class in same way. We will have a one dimensional dynamic array as private data member of our class that will store all the vectors in row major order. Note that you cannot use 2 dimensional static or dynamic arrays here. You would also need to determine other data members that your class needs. Make sure you don't make extra/redundant variables but only the ones that are necessary. *VectorOfVectors* will have the following interface at least (i.e. you might need to add other functions):

```
public:
bool insert(int value, int vectorNo);                   //inserts the value in vectorNo
                                                        //mentioned
bool remove(int vectorNo, int index);                   //removes the value from the index
                                                        //of the vector no mentioned
bool change(int vectorNo, int index, int newValue);     //changes the value of the index
                                                        //of the mentioned vector no.
void printAll();                                        //prints all values in 2D format
bool get(int vectorNo, int index, int &value);          //returns the value at index in
                                                        //vector No in value variable
void removefromVectors(int value);                      //removes the row and column
                                                        //containing the value and
                                                        //compacts it
```

`removefromVectors(15)` **Example :**
The empty cells can be filled with -1.

2	3	5	6	10	22
3	3	12	15	20	21
6	8	10	11	30	34
2	7	9	10	13	15

2	3	5	10	-	-
6	7	10	30	-	-
-	-	-	-	-	-
-	-	-	-	-	-

Note that each vector in the object stores non-negative integers arranged in ascending order. Whenever we need to insert an element and the required vector is out of space, we increase the size of all vectors by ½ of the original size. So in a way vectors are symmetric.

You are required to use visual studio (version 10 or above) c++ for this assignment. Make separate .cpp and .h files.

Q # 2

Implement template based stack.

Q # 3

Implement an algorithm to convert a postfix expression to infix form that includes parenthesis, plus, minus, division and multiplication operator.