NAME: SATVIK DANDALE  
DIV: TY-E

GR NO: 1710797

ROLL NO: 13

BATCH: B1

/\*

    Implement your own Vector class using templates in C++. The new Vector should have the following functionalities:

        1. void add(int element): Given an element, add it to the Vector (append)

        2. int del(int index): Given a position (index), delete the element at that position.

        3. void modify(int index, int element): At the given index position, replace the element with the given element.

        4. void display(): Display the complete Vector.

        5. void multiply(int scalar): Multiply all the elements in the Vector with the given scalar value.

\*/

#include<bits/stdc++.h>

using namespace std;

template <class T>

class Vector{

    private:

        T \*array;   // Vector is a collection. Array to store the elements.

        int size;

    public:

        Vector(){   // Size not given. NULL vector.

            size = 0;

            array = NULL;

        }

        Vector(int size){   // Size is given

            this->size = size;

            array = new T[size];

        }

        void add(T element){

            /\*

                STEPS:

                    1. Create a temp array of size+1

                    2. Copy origianl array to temp

                    3. Add new element to temp array at the last

                    4. Delete original array's memory

                    5. Increment this->size by 1

                    6. Create new array of size which will be pointed by original array

                    7. Copy temp to original array

                    8. Delete temp's memory

            \*/

            T \*temp = new T[this->size+1];  // STEP 1

            for(int i=0; i < this->size; i++)   // STEP 2

                temp[i] = this->array[i];

            temp[size] = element;   // STEP 3

            delete[] this->array;   // STEP 4

            this->size += 1;    // STEP 5

            array = new T[size];    // STEP 6

            for(int i=0; i < this->size; i++)   // STEP 7

                this->array[i] = temp[i];

            delete[] temp;  // STEP 8

        }

        int del(int index){

            // CHECK IF THE INDEX IS LESS THAN SIZE

            if (index >= size){

                throw "Index out of bound!";

                return 0;

            }

            /\*

                1. Create a temp array pointing to original array.

                2. Assign new memory of size -1 to original array.

                3. Copy all elements from temp to original except the index given.

                4. Decrement size by 1

                5. Save the element of the given index

                6. Delete the memory pointed by temp

                7. Return the saved element.

            \*/

            T \*temp = this->array;

            this->array = new T[this->size-1];

            for(int i=0; i < this->size; i++)

                if(i != index)

                    this->array[i] = temp[i];

            this->size -= 1;

            int element = temp[index];

            delete[] temp;

            return element;

        }

        void modify(int index, int element){

            // Change the element given with the index by element

            if (index >= size){

                throw "Index out of bound!";

                return;

            }

            this->array[index] = element;

        }

        void display(){

            cout<<"CURRENT VECTOR:\n";

            for(int i=0; i < this->size; i++)

                cout<<this->array[i]<<"\t";

            cout<<endl;

        }

        void multiply(int scalar){

            // Multiple all elements with scalar quantity

            for(int i=0; i < this->size; i++)

                this->array[i] \*= scalar;

        }

        void multiply(float scalar){

            // Multiple all elements with scalar quantity

            for(int i=0; i < this->size; i++)

                this->array[i] \*= scalar;

        }

};

int main(){

    Vector<char> v;

    v.add('a');

    v.add('c');

    v.add('b');

    v.del(2);

    v.display();

}