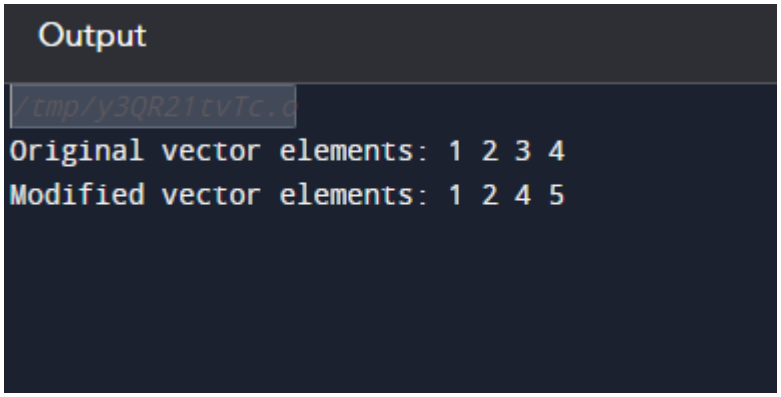
NAME: Haider Nawaz Cheema
CMS 480239
ME 15 C

Q 01 Iterate Through Vector Using Iterators and print all pushed elements. Next you need to push integer 5 and remove element at that position

CODE:

```
#include <iostream>
#include <vector>
using namespace std;

int main() {
    vector<int> v = {1, 2, 3, 4};

    cout << "Original vector elements: ";
    for (auto it = v.begin(); it != v.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl;

    v.push_back(5);

    int r = 2;
    if (r >= 0 && r < v.size()) {
        v.erase(v.begin() + r);
    }

    cout << "Modified vector elements: ";
    for (auto it = v.begin(); it != v.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl;

    return 0;
}
```

RESULT:



Q 02 Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int)

a. Ask the user for the number of name/grade pairs that will be entered.
b. Display the mean of the grades.
c. Display the median of the grades.
d. Display the mode of the grades.
e. Display the names of the students with the mode as their grade.

CODE:

```
#include <bits/stdc++.h>
using namespace std;

int main() {
    int numPairs;
    cout << "Enter the number of name/grade pairs: ";
    cin >> numPairs;

    vector<string> studentNames;
    vector<int> studentGrades;

    for (int i = 0; i < numPairs; ++i) {
        string name;
        int grade;

        cout << "Enter name #" << i + 1 << ": ";
        cin >> name;
```

```
        cout << "Enter grade for " << name << ": ";
        cin >> grade;

        studentNames.push_back(name);
        studentGrades.push_back(grade);
    }

    double mean = accumulate(studentGrades.begin(), studentGrades.end(), 0.0) /
numPairs;
    cout << "Mean of grades: " << fixed << setprecision(2) << mean << endl;

    sort(studentGrades.begin(), studentGrades.end());
    int medianIndex = numPairs / 2;
    double median;

    if (numPairs % 2 == 0) {
        median = (studentGrades[medianIndex - 1] +
studentGrades[medianIndex]) / 2.0;
    } else {
        median = studentGrades[medianIndex];
    }

    cout << "Median of grades: " << fixed << setprecision(2) << median << endl;

    unordered_map<int, int> gradeFrequency;
    int maxFrequency = 0;
    int mode;

    for (int grade : studentGrades) {
        gradeFrequency[grade]++;
        if (gradeFrequency[grade] > maxFrequency) {
            maxFrequency = gradeFrequency[grade];
            mode = grade;
        }
    }

    cout << "Mode of grades: " << mode << " (occurs " << maxFrequency << "
times)" << endl;

    cout << "Names of students with the mode grade (" << mode << "): ";
    for (int i = 0; i < numPairs; ++i) {
        if (studentGrades[i] == mode) {
```

```
            cout << studentNames[i] << " ";
        }
    }
    cout << endl;

    return 0;
}
```

RESULT: