Shah Jahan khan (469192)

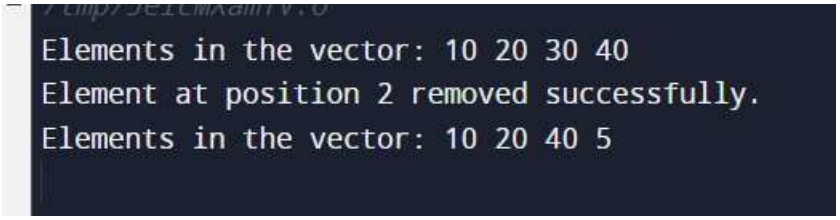
ME-15 section (C)

Lab manual 10

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

```
#include <iostream>
#include <vector>
using namespace std;
void printVector(const vector<int>& vec) {
   cout << "Elements in the vector: ";
   for (int i = 0; i < vec.size(); ++i) {
      cout << vec[i] << " ";
   }
   cout << endl;
}

int main () {
   vector<int> vec;

   vec.push_back(10);
   vec.push_back(20);
   vec.push_back(30);
   vec.push_back(40);

   printVector(vec);
   vec.push_back(5);
   int positionToRemove = 2;
```

```
    if (positionToRemove >= 0 && positionToRemove < vec.size()) {
        vec.erase(vec.begin() + positionToRemove);
        cout << "Element at position " << positionToRemove << " removed successfully." <<endl;
    } else {
        cout << "Invalid position to remove." << endl;
    }

    printVector(vec);

    return 0;
}
```



2. 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.

Display the names of the students with the mode as their grade.

```
#include <iostream>
#include <vector>
#include <string>
#include <algorithm>
#include <unordered_map>
#include<numeric>
using namespace std;

int main() {
```

```
int numPairs;
cout << "Enter the number of name/grade pairs: ";
cin >> numPairs;

vector<string> names(numPairs);
vector<int> grades(numPairs);

cout << "Enter names and grades:" << endl;
for (int i = 0; i < numPairs; ++i) {
   cout << "Name " << i + 1 << ": ";
   cin >> names[i];
   cout << "Grade " << i + 1 << ": ";
   cin >> grades[i];
}

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



sort(grades.begin(), grades.end());
double median;
if (numPairs % 2 == 0) {
   median = (grades[numPairs / 2 - 1] + grades[numPairs / 2]) / 2.0;
} else {
   median = grades[numPairs / 2];
}
cout << "Median of the grades: " << median << endl;

unordered_map<int, int> freq;
int mode = -1, maxFreq = 0;
for (int grade : grades) {
```

```
        maxFreq = max(maxFreq, ++freq[grade]);
        if (freq[grade] == maxFreq) {
            mode = grade;
        }
    }
    cout << "Mode of the grades: " << mode << endl;

    cout << "Names of students with the mode as their grade:" << endl;
    for (int i = 0; i < numPairs; ++i) {
        if (grades[i] == mode) {
            cout << names[i] << endl;
        }
    }

    return 0;
}
```