

National University of Sciences and Technology (NUST)

Department of Mechanical Engineering (SMME)



Fundamentals of Programming (FOP)

[Home Tasks](#)

[Lab Manual 10](#)

By

Muhammad Owais

461359

Teacher: Sir Muhammad Affan

Task-1:

```
#include <iostream>
#include <vector>
using namespace std;
int main() {
    vector<int> V;
    V.push_back(1);
    V.push_back(2);
    V.push_back(3);
    V.push_back(4);
    cout << "Original elements in the vector: ";
    for (vector<int>::iterator it = V.begin(); it != V.end(); ++it) {
        cout << *it << " ";
    }
    cout<<endl;
    V.push_back(5);
    cout << "Elements after pushing 5: ";
    for (vector<int>::iterator it = V.begin(); it != V.end(); ++it) {
        cout << *it << " ";
    }
    cout<<endl;
    int positionToRemove = 2;
    if (positionToRemove >= 0 && positionToRemove < V.size()) {
        vector<int>::iterator itToRemove = V.begin() + positionToRemove;
        V.erase(itToRemove);
    } else {
        cout << "Invalid position to remove." <<endl;
    }
    cout << "Elements after removing element at position " << positionToRemove << ": ";
    for (vector<int>::iterator it = V.begin(); it != V.end(); ++it) {
        cout << *it << " ";
    }
    cout<<endl;
    return 0;
}
```

```
Original elements in the vector: 1 2 3 4
Elements after pushing 5: 1 2 3 4 5
Elements after removing element at position 2: 1 2 4 5

-----
Process exited after 0.03531 seconds with return value 0
Press any key to continue . . .
```

Task-2:

```
#include <iostream>
#include <vector>
#include <algorithm>
#include <map>
#include <numeric>
#include <iomanip>
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);
    for (int i = 0; i < numPairs; ++i) {
        cout << "Enter name for student " << i + 1 << ": "; cin >> names[i];
        cout << "Enter grade for student " << i + 1 << ": "; cin >> grades[i];
    }
    double sum = accumulate(grades.begin(), grades.end(), 0); double mean = sum / numPairs;
    cout << "Mean of the grades: " << fixed << setprecision(2) << mean << endl;
    sort(grades.begin(), grades.end());
    double median = (numPairs % 2 == 0) ? (grades[numPairs / 2 - 1] + grades[numPairs / 2]) / 2.0 : grades[numPairs / 2];
    cout << "Median of the grades: " << fixed << setprecision(2) << median << endl;
    map<int, int> gradeCount;
    for (int i = 0; i < numPairs; ++i) {
        gradeCount[grades[i]]++;
    }
    int mode = grades[0];
    int maxCount = gradeCount[grades[0]];
    for (int i = 1; i < numPairs; ++i) {
        if (gradeCount[grades[i]] > maxCount) {
            mode = grades[i];
            maxCount = gradeCount[grades[i]];
        }
    }
    cout << "Mode of the grades: " << mode << endl; cout << "Names of students with the mode grade (" << mode << "):" << endl;
    for (int i = 0; i < numPairs; ++i) {
        if (grades[i] == mode) {
            cout << names[i] << endl;
        }
    }
    return 0;
}
```

```
Enter the number of name/grade pairs: 3
Enter name for student 1: max
Enter grade for student 1: 60
Enter name for student 2: fin
Enter grade for student 2: 70
Enter name for student 3: eggster
Enter grade for student 3: 100
Mean of the grades: 76.67
Median of the grades: 70.00
Mode of the grades: 60
Names of students with the mode grade (60):
max
-----
Process exited after 20.33 seconds with return value 0
Press any key to continue . . .
```