

**Fundamentals of**

**Programming**

**Hunzala Asaad**

**474644**

**Sir Affan tariq**

**Task 1:**

**#include<iostream>**

**#include<vector>**

**using namespace std;**

**int main(){**

**int n=0;**

**cout<<"enter the number of elements in your vector: ";**

**cin>>n;**

**vector<int>v(n);**

**for(int i=0;i<v.size();i++){**

**cin>>v[i];**

**}**

**vector<int>::iterator it;**

**for(it=v.begin();it!=v.end();it++){**

**cout<<\*it<<" ";**

**}**

**cout<<endl;**

**v.push\_back(5);**

**for(it=v.begin();it!=v.end();it++){**

**cout<<\*it<<" ";**

**}**

**cout<<endl;**

**v.pop\_back();**

**for(it=v.begin();it!=v.end();it++){**

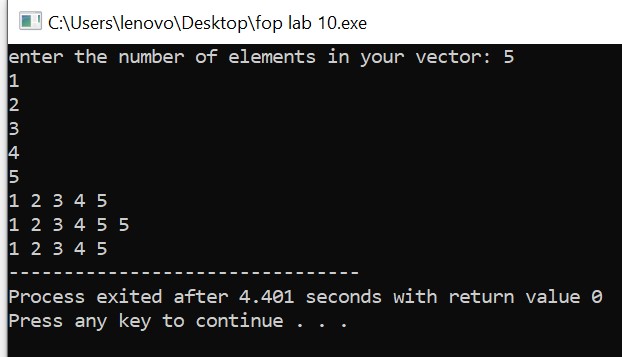
**cout<<\*it<<" ";**

**}**

**return 0;**

**}**

**Output:**

****

**Task 2:**

**#include <iostream>**

**#include <vector>**

**#include <algorithm>**

**#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);**

**for (int i = 0; i < numPairs; i++) {**

**cout << "Enter name of student" << i + 1 << ": ";**

**cin >> names[i];**

**cout << "Enter grade of " << names[i] << ": ";**

**cin >> grades[i];**

**}**

**double mean = accumulate(grades.begin(), grades.end(), 0.0) / numPairs;**

**cout << "Mean grade: " << 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 grade: " << median << endl;**

**int modeCount = 1, maxModeCount = 1;**

**int mode = grades[0];**

**for (int i = 1; i < numPairs; i++) {**

**if (grades[i] == grades[i - 1]) {**

**modeCount++;**

**} else {**

**modeCount = 1;**

**}**

**if (modeCount > maxModeCount) {**

**maxModeCount = modeCount;**

**mode = grades[i];**

**}**

**}**

**cout << "Mode grade: " << mode << endl;**

**cout << "name of student with mode grade: ";**

**for (int i = 0; i < numPairs; i++) {**

**if (grades[i] == mode) {**

**cout << names[i] << " ";**

**}**

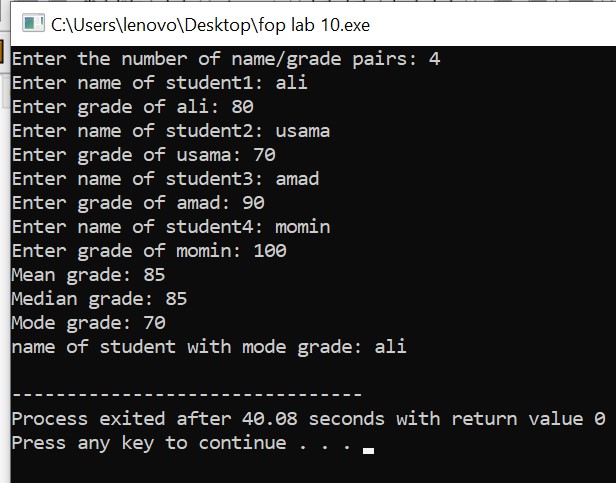
**}**

**cout << endl;**

**return 0;**

**}**

**Output:**

****