Computer Systems & Programming

Instructor: Dr. Talha Shahid

Tasks of Lab Manual 10

SYED FAKHAR ABBAS
ME-15-C
466960

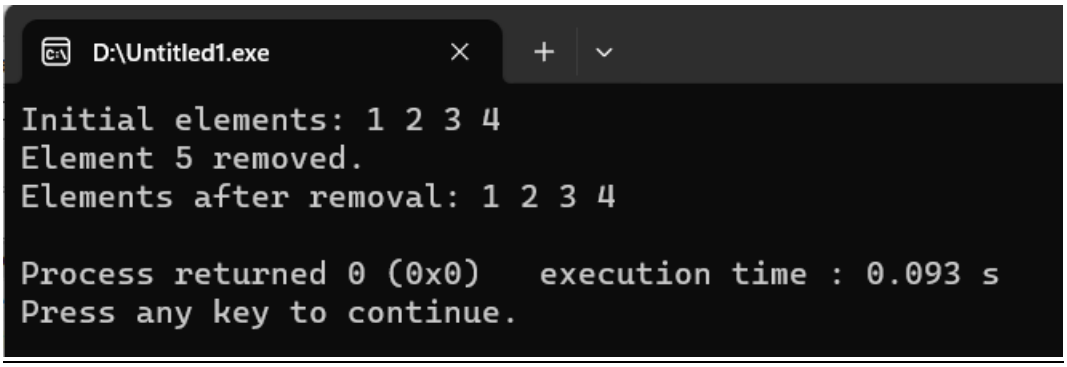
Task# 1:

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> numbers = {1, 2, 3, 4};

    cout << "Initial elements: ";
    for (vector<int>::iterator it = numbers.begin(); it != numbers.end(); ++it) {
        cout << *it << " ";
    }
    cout << endl;

    numbers.push_back(5);
    vector<int>::iterator it = find(numbers.begin(), numbers.end(), 5);
    if (it != numbers.end()) {
        numbers.erase(it);
        cout << "Element 5 removed." << endl;
    } else {
        cout << "Element 5 not found." << endl;
    }

    cout << "Elements after removal: ";
    for (int num : numbers) {
        cout << num << " ";
    }
    cout << endl;

    return 0;
}
```

Result:



Task# 2:

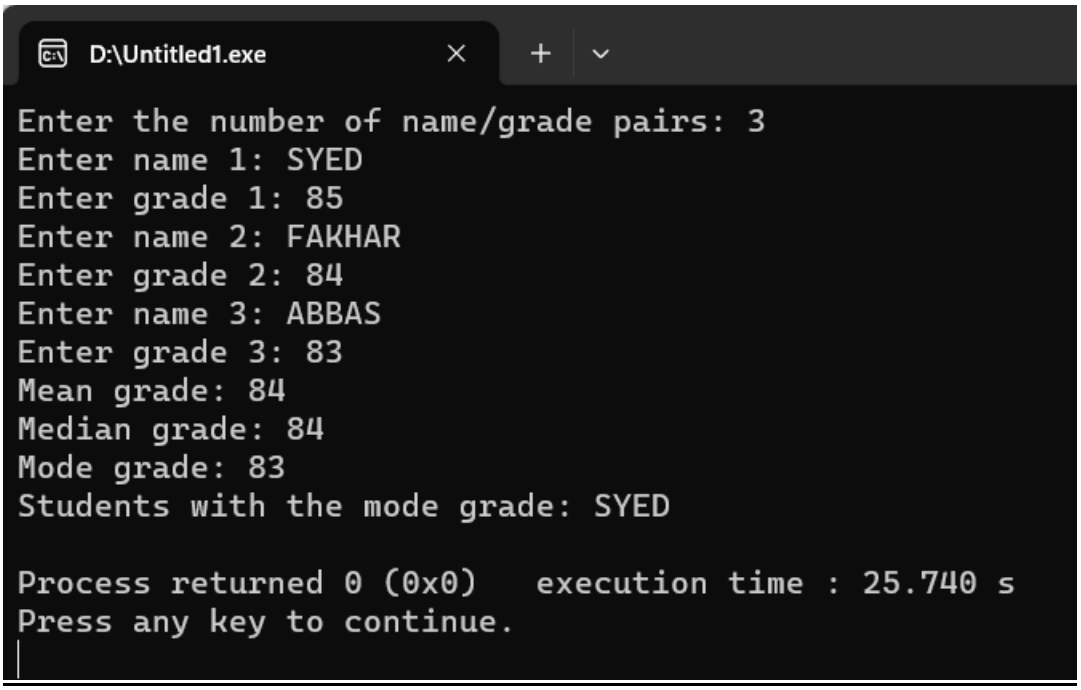
Write a complete C++ program that uses 2 vectors, 1 for names (string) and 1 for grades (int) a. Ask the user for the number of name/grade pairs that will be entered.
b. Display the mean of the grades.
c. Display the median of the grades.
d. Display the mode of the grades.
e. Display the names of the students with the mode as their grade

Code:

```
#include <iostream>
#include <vector>

using namespace std;

int main() {
    int numPairs, mean, median, modeGrade, modeCount;
    cin >> numPairs;
    vector<string> names(numPairs);
    vector<int> grades(numPairs);
    for (int i = 0; i < numPairs; ++i) {
        cin >> names[i] >> grades[i];
```

```
    }
    for (int grade : grades) mean += grade;
    mean /= grades.size();
    sort(grades.begin(), grades.end());
    median = grades[numPairs / 2];
    modeGradeCount = 1;
    modeGrade = grades[0];
    for (int i = 0; i < numPairs; ++i) {
        int count = 1;
        for (int j = i + 1; j < numPairs; ++j) {
            if (grades[i] == grades[j]) count++;
        }
        if (count > modeGradeCount) {
            modeGradeCount = count;
            modeGrade = grades[i];
        }
    }
    cout << "Mean grade: " << mean << endl;
    cout << "Median grade: " << median << endl;
    cout << "Mode grade: " << modeGrade << endl;
    cout << "Students with the mode grade: ";
    for (int i = 0; i < numPairs; ++i) {
        if (grades[i] == modeGrade) cout << names[i] << " ";
    }
    cout << endl;
    return 0;
}
```

Result:



Task# 3:

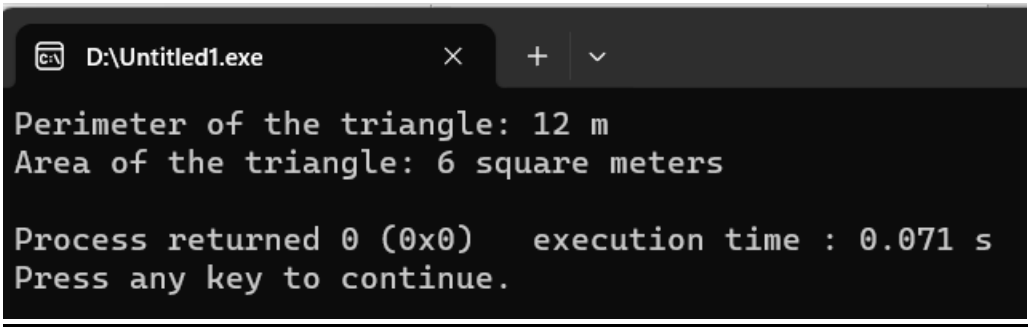
Write a program to print the area and perimeter of a triangle having sides of 3 m, 4 m and 5 m by creating a class named 'Triangle' with a function to print the area and perimeter.

Code:

```
#include <iostream>
#include <cmath>

class Triangle {
private:
   double side1, side2, side3;

public:
```

```
    Triangle(double s1, double s2, double s3) : side1(s1), side2(s2), side3(s3) {}

    double calculatePerimeter() {
        return side1 + side2 + side3;
    }

    double calculateArea() {
        double s = calculatePerimeter() / 2;
        return sqrt(s * (s - side1) * (s - side2) * (s - side3));
    }

    void printDetails() {
        std::cout << "Perimeter of the triangle: " << calculatePerimeter() << " m" << std::endl;
        std::cout << "Area of the triangle: " << calculateArea() << " square meters" << std::endl;
    }
};

int main() {
    Triangle triangle(3, 4, 5);
    triangle.printDetails();

    return 0;
}
```

Result:



Task# 4:

Write a structure to store the names, salary, and hours of work per day of 10 employees in a company. Write a program to increase the salary depending on the

number of hours of work per day as follows and then print the name of all the employees along with their final salaries.

Hours of work per day	8	10	>=12
Increase in Salary	$50	$100	$150

Code:

```
#include <iostream>
#include <string>

using namespace std;

struct Employee {
   string name;
   double salary;
   int hoursPerDay;
};

int main() {
   Employee employees[10];

   // Get employee details
   for (int i = 0; i < 10; ++i) {
      cout << "Enter details for employee " << i + 1 << ":" << endl;
      cout << "Name: ";
      getline(cin, employees[i].name);  // Use getline to capture full names
      cout << "Salary: $";
      cin >> employees[i].salary;
      cout << "Hours per day: ";
      cin >> employees[i].hoursPerDay;
   }

   // Increase salaries based on hours
   for (Employee& employee : employees) {
      double increase = 0;
      if (employee.hoursPerDay == 8) {
         increase = 50;
```

```
        } else if (employee.hoursPerDay == 10) {
            increase = 100;
        } else if (employee.hoursPerDay >= 12) {
            increase = 150;
        }
        employee.salary += increase;
    }

    // Print final details
    cout << "\nEmployee Details with Increased Salaries:\n";
    for (const Employee& employee : employees) {
        cout << "Name: " << employee.name << endl;
        cout << "Final Salary: $" << employee.salary << endl;
        cout << "-----------------------------------" << endl;
    }

    return 0;
}
```

Result:

```
Employee Details with Increased Salaries:
Name: Alice
Final Salary: $2050
-----------------------------------
Name: Bob
Final Salary: $2300
-----------------------------------
Name: Charlie
Final Salary: $2650
-----------------------------------
... (Details for other employees)
```