TASKS OF LAB MANUAL 10

By

MUHAMMAD AHMED ME-15-C

(455401)

SUBMITTED TO: SIR AFFAN

SUBJECT: FOP

DATE: 28/12/23

Task# 1:

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;
}
```

Task# 2:

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;
}
```

Task# 3:

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;
}
```

Task# 4:

Code:

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

using namespace std;

struct Employee {
    string name;
```

```
    double salary;
    int hoursPerDay;
};

int main() {
    Employee employees[10];

    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;
    }
    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;
    }
    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;
}
```