Fadil Faisal 24K-0635 (L1)

# Q1

*/\**

*Q1: Write a C++ program to check whether a given number is prime or not. Allow*

*the user to input a number and display whether it's prime or not?*

*\*/*

#include <iostream>

using namespace std;

bool is\_prime(int *x*)

{

    if (*x* <= 1)

        return false;

    for (int i = 2; i < *x*; i++)

    {

        if (*x* % i == 0)

            return false;

    }

    return true;

}

int main()

{

    int x;

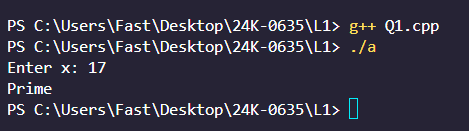
    cout << "Enter x: ";

    cin >> x;

    cout << (is\_prime(x) ? "Prime" : "Not prime");

    return 0;

}



# Q2

*/\**

*Q2: Design a C++ program to manage student marks. Allow the user to input*

*marks for students in three subjects (Mathematics, English, and Science). The*

*program should calculate the total marks, average marks, and display the grade*

*for each student. The user can specify the number of students and then input*

*the marks for each subject for each student. Finally, display the marks, total,*

*average, and grade for each student. Assume a grading system with the*

*following criteria:*

*90 or above: Grade A*

*80-89: Grade B*

*70-79: Grade C*

*60-69: Grade D*

*Below 60: Grade F*

*\*/*

#include <iostream>

using namespace std;

typedef struct

{

    int maths;

    int eng;

    int sci;

} Student;

char calc\_grade(int *marks*)

{

    if (*marks* >= 90)

        return 'A';

    else if (*marks* >= 80)

        return 'B';

    else if (*marks* >= 70)

        return 'C';

    else if (*marks* >= 60)

        return 'D';

    else

        return 'F';

}

int main()

{

    int n;

    cout << "Enter num of students: ";

    cin >> n;

    Student \*students = new Student[n];

    cout << "\t==== STUDENT INPUT ====\n";

    for (int i = 0; i < n; i++)

    {

        cout << "STUDENT " << i + 1 << "\n";

        cout << "Enter marks (Maths): ";

        cin >> students[i].maths;

        cout << "Enter marks (English): ";

        cin >> students[i].eng;

        cout << "Enter marks (Science): ";

        cin >> students[i].sci;

        cout << "\n";

    }

    cout << "\t==== STUDENT RESULTS ====\n";

    for (int i = 0; i < n; i++)

    {

        cout << "STUDENT " << i + 1 << "\n";

        cout << "Maths: " << calc\_grade(students[i].maths) << "\n";

        cout << "English: " << calc\_grade(students[i].eng) << "\n";

        cout << "Science: " << calc\_grade(students[i].sci) << "\n";

        int total = students[i].maths + students[i].eng + students[i].sci;

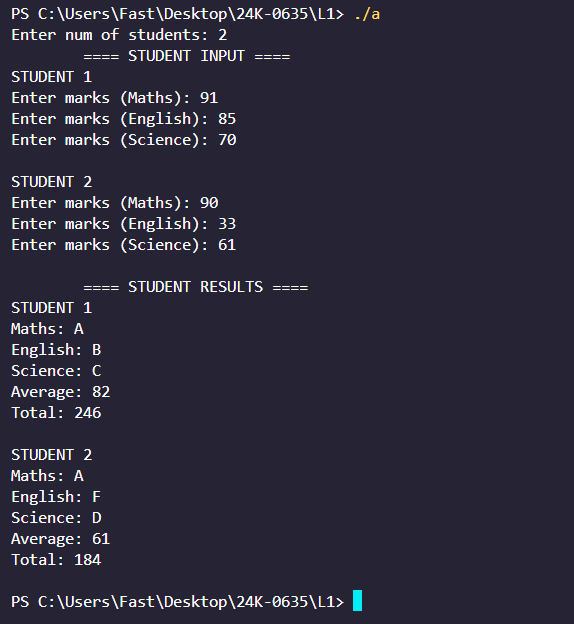
        cout << "Average: " << total / 3 << "\n";

        cout << "Total: " << total << "\n";

        cout << "\n";

    }

}



# Q3

*/\**

*Q3: Given an array of integers nums and an integer target, return indices of the*

*two numbers such that they add up to target.*

*You may assume that each input would have exactly one solution, and you may*

*not use the same element twice.You can return the answer in any order.*

*\*/*

#include <iostream>

using namespace std;

int main()

{

    int n;

    int target;

    cout << "Enter size: ";

    cin >> n;

    cout << "Enter target: ";

    cin >> target;

    int \*nums = new int[n];

    for (int i = 0; i < n; i++)

    {

        cout << "Enter element " << i << ": ";

        cin >> nums[i];

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            if (nums[i] + nums[j] == target)

            {

                cout << "[" << i << ", " << j << "]";

                return 0;

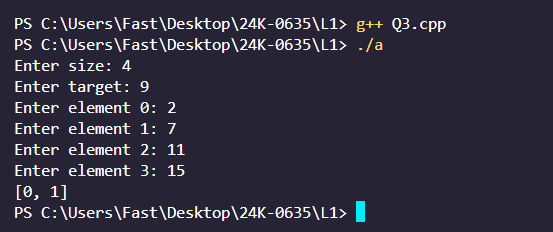
            }

        }

    }

    cout << "No pair found";

}



# Q4

#include <iostream>

using namespace std;

*// arr(9)=[0,1,2,3,4,5,6,7,8];*

*// xlength=j-i*

*// if arr[j]<arr[i]*

*// ylength=arr[j]*

*// else*

*// ylength=arr[i]*

*// area= xlength \* ylength*

int main()

{

    int n;

    int h, d, area = 0;

    int max\_area = 0;

    cout << "Enter n: ";

    cin >> n;

    int \*heights = new int[n];

    for (int i = 0; i < n; i++)

    {

        cout << "Enter element " << i << ": ";

        cin >> heights[i];

    }

    for (int i = 0; i < n; i++)

    {

        for (int j = i + 1; j < n; j++)

        {

            d = j - i;

            if (heights[i] < heights[j])

                h = heights[i];

            else

                h = heights[j];

            area = d \* h;

            if (area > max\_area)

                max\_area = area;

        }

    }

    cout << "Max Area: " << max\_area;

    return 0;

}

