**BUSINESS LOGIC**

using System;

using System.Collections.Generic;

using System.Linq;

using System.Runtime.ConstrainedExecution;

using System.Security.Cryptography.X509Certificates;

using System.Text;

using System.Threading.Tasks;

namespace assignment

{

public class completecall

{

internal void quadrant(int x, int y)

{

int x1, y1;

x1 = x;

y1 = y;

if (x1 > 0 && y1 > 0)

{

Console.WriteLine(" The Coordinate point (" + x1 + "," + y1 + ") lies in the First quadrant");

}

else if (x1 < 0 && y1 < 0)

{

Console.WriteLine(" The Coordinate point (" + x1 + "," + y1 + ") lies in the Third quadrant");

}

else if (x1 > 0 && y1 < 0)

{

Console.WriteLine(" The Coordinate point (" + x1 + "," + y1 + ") lies in the Fourth quadrant");

}

else if (x1 == 0 && y1 == 0)

{

Console.WriteLine(" The Coordinate point (" + x1 + "," + y1 + ") lies in the same orgin");

}

else

{

Console.WriteLine(" The Coordinate point (" + x1 + "," + y1 + ") lies in the Second quadrant");

}

}

internal void student\_details(int roll, string name, int m1, int m2, int m3)

{

int total, roll1, mark1, mark2, mark3;

decimal percentage;

string sname, Division;

roll1 = roll;

sname = name;

mark1 = m1;

mark2 = m2;

mark3 = m3;

total = mark1 + mark2 + mark3;

percentage = total / 3;

if (percentage <= 80)

{

Division = "First";

}

else if (percentage <= 60)

{

Division = "Second";

}

else

{

Division = "Third";

}

Console.WriteLine("Roll No : " + roll1);

Console.WriteLine("Name of the Student :" + sname);

Console.WriteLine("Marks in Physics :" + mark1);

Console.WriteLine("Marks in Chemistry :" + mark2);

Console.WriteLine("Marks in Computer Application :" + mark3);

Console.WriteLine("Total Marks :" + total);

Console.WriteLine("Percentage :" + percentage);

Console.WriteLine("Division :" + Division);

}

internal void triangle1()

{

int i, j, number = 1;

for (i = 1; i < 5; i++)

{

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

{

Console.Write(number);

++number;

}

Console.WriteLine();

}

}

internal void triangel2(int row)

{

int i, j, number = 1, space, rows;

rows = row;

space = rows + 4 - 1;

for (i = 1; i <= rows; i++)

{

for (j = space; j >= 1; j--)

{

Console.Write(" ");

}

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

Console.Write(" " + number++);

Console.Write("\n");

space--;

}

}

internal void electricity()

{

int custid, consumed\_unit;

double charge, surchg = 0, gramt, netamt;

string name;

Console.Write("Input Customer ID :");

custid = Convert.ToInt32(Console.ReadLine());

Console.Write("Input the name of the customer :");

name = Console.ReadLine();

Console.Write("Input the unit consumed by the customer : ");

consumed\_unit = Convert.ToInt32(Console.ReadLine());

if (consumed\_unit < 200)

charge = 1.20;

else if (consumed\_unit >= 200 && consumed\_unit < 400)

charge = 1.50;

else if (consumed\_unit >= 400 && consumed\_unit < 600)

charge = 1.80;

else

charge = 2.00;

gramt = consumed\_unit \* charge;

if (gramt > 400)

surchg = gramt \* 15 / 100;

netamt = gramt + surchg;

if (netamt < 100)

netamt = 100;

Console.WriteLine("Customer IDNO " + custid);

Console.WriteLine("Customer Name " + name);

Console.WriteLine("unit Consumed " + consumed\_unit);

Console.WriteLine("Amount Charges @Rs." + charge);

Console.WriteLine("Surchage Amount " + surchg);

Console.WriteLine("Net Amount Paid By the Customer " + netamt);

}

internal void reversearray()

{

int[] arr;

int n;

n = Convert.ToInt32(Console.ReadLine());

arr = new int[n];

for (int i = 0; i < n; i++)

arr[i] = Convert.ToInt32(Console.ReadLine());

for (int i = n - 1; i >= 0; i--)

Console.Write("" + arr[i]);

}

internal void copyarray()

{

int[] arr;

int[] arr1;

int n, i, j;

n = Convert.ToInt32(Console.ReadLine());

arr = new int[n];

arr1 = new int[n];

for (i = 0; i < n; i++)

arr[i] = Convert.ToInt32(Console.ReadLine());

for (i = 0; i < n; i++)

arr1[i] = arr[i];

;

for (i = 0; i < n; i++)

Console.Write(arr1[i]);

}

internal void sumarray()

{

int[] arr = { 10, 20, 30, 40, 50 };

int total = 0;

for (int i = 0; i < arr.Length; i++)

total = total + arr[i];

Console.WriteLine(total);

}

internal void uniques()

{

int n, ctr = 0;

int[] arr1 = new int[100];

int i, j, k;

Console.Write("Input the number of elements to be stored in the array :");

n = Convert.ToInt32(Console.ReadLine());

Console.Write("Input {0} elements in the array :\n", n);

for (i = 0; i < n; i++)

{

Console.Write("element - {0} : ", i);

arr1[i] = Convert.ToInt32(Console.ReadLine());

}

/\*Checking duplicate elements in the array \*/

Console.Write("\nThe unique elements found in the array are : \n");

for (i = 0; i < n; i++)

{

ctr = 0;

for (j = 0; j < i - 1; j++)

{

if (arr1[i] == arr1[j])

{

ctr++;

}

}

for (k = i + 1; k < n; k++)

{

if (arr1[i] == arr1[k])

{

ctr++;

}

if (arr1[i] == arr1[i + 1])

{

i++;

}

}

if (ctr == 0)

{

Console.Write("{0} ", arr1[i]);

}

}

}

internal void secondlargest()

{

{

int[] array = { 2, 11, 15, 1, 7, 99, 6, 85, 4 };

Array.Sort(array);

Array.Reverse(array);

Console.WriteLine("Second Highest Value In Array " + array[1]);

foreach (var result in array)

{

Console.Write(result + " ");

}

}

}

internal void transpose()

{

int[,] matrix = new int[,] { { 1, 4, 7 }, { 2, 5, 8 }, { 3, 6, 9 } };

int rows = matrix.GetLength(0);

int cols = matrix.GetLength(1);

int[,] transposedMatrix = new int[cols, rows];

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

transposedMatrix[j, i] = matrix[i, j];

}

}

// Display the original matrix

Console.WriteLine("Original Matrix:");

for (int i = 0; i < rows; i++)

{

for (int j = 0; j < cols; j++)

{

Console.Write(matrix[i, j] + " ");

}

Console.WriteLine();

}

// Display the transposed matrix

Console.WriteLine("Transposed Matrix:");

for (int i = 0; i < cols; i++)

{

for (int j = 0; j < rows; j++)

{

Console.Write(transposedMatrix[i, j] + " ");

}

}

}

internal void prime()

{

int[] arr = new int[] { 1, 2, 3, 4, 5 };

int sum = 0;

bool IsPrime(int num)

{

if (num <= 1)

{

return false;

}

for (int i = 2; i <= Math.Sqrt(num); i++)

{

if (num % i == 0)

{

return false;

}

}

return true;

}

foreach (int num in arr)

{

if (IsPrime(num))

{

sum += num;

}

}

Console.WriteLine("The sum of the prime numbers in the array is: {0}", sum);

}

internal void countstring()

{

string str = "This is a sample string with multiple words.";

string[] words = str.Split(new char[] { ' ' }, StringSplitOptions.RemoveEmptyEntries);

int wordCount = words.Length;

Console.WriteLine("The total number of words in the string is: " + wordCount);

}

internal void totalcount()

{

string str = "Different Char@cter& 12345";

int alphabetCount = 0;

int digitCount = 0;

int specialCount = 0;

foreach (char c in str)

{

if (Char.IsLetter(c))

{

alphabetCount++;

}

else if (Char.IsDigit(c))

{

digitCount++;

}

else if (!Char.IsWhiteSpace(c))

{

specialCount++;

}

}

Console.WriteLine("The total number of alphabets in the string is: " + alphabetCount);

Console.WriteLine("The total number of digits in the string is: " + digitCount);

Console.WriteLine("The total number of special characters in the string is: " + specialCount);

}

internal void fibo(int n)

{

int n1 = 0, n2 = 1, n3;

if (n == 0)

return;

Console.Write(n1 + " " + n2 + " ");

for (int i = 2; i < n; ++i)

{

n3 = n1 + n2;

Console.Write(n3 + " ");

n1 = n2;

n2 = n3;

}

}

internal void armstrong(int num)

{

while (num != 0)

{

int sum = 0;

int temp = num;

while (temp > 0)

{

int digit = temp % 10;

sum += digit \* digit \* digit;

temp /= 10;

}

if (num == sum)

{

Console.WriteLine("Yes! it's Armstrong number");

}

else

{

Console.WriteLine("No! it's Armstrong number");

}

}

}

internal void primes(int num)

{

if (num <= 0)

{

Console.WriteLine("Enter the possitive number");

}

else

{

if (num % 2 == 0)

{

Console.WriteLine("Yes.It's prime number");

}

else

{

Console.WriteLine("No,It's not a prime number");

}

}

}

internal void stringpalindrome(string word)

{

bool isPalindrome = true;

int i = 0;

int j = word.Length - 1;

while (i < j)

{

if (word[i] != word[j])

{

isPalindrome = false;

break;

}

i++;

j--;

}

if (isPalindrome)

{

Console.WriteLine("The string is a palindrome.");

}

else

{

Console.WriteLine("The string is not a palindrome.");

}

}

internal void numpalindrome(int num)

{

int reversedNumber = 0, remainder, originalNumber = num;

while (num > 0)

{

remainder = num % 10;

reversedNumber = (reversedNumber \* 10) + remainder;

num /= 10;

}

if (originalNumber == reversedNumber)

{

Console.WriteLine("The number is a palindrome.");

}

else

{

Console.WriteLine("The number is not a palindrome.");

}

}

}

}

// See https://aka.ms/new-console-template for more information

using assignment;

using System;

class Program

{

public static void Main(string[] args)

{

completecall call = new completecall();

//Quadrant

int xaxis, yaxis;

Console.WriteLine("The Value of X coordinate :");

xaxis = Convert.ToInt32(Console.ReadLine());

Console.WriteLine("The Value of Y coordinate :");

yaxis = Convert.ToInt32(Console.ReadLine());

call.quadrant(xaxis, yaxis);

// Student Details

int rollno, phy\_mark, chem\_mark, com\_mark, total\_mark;

decimal percentage;

string division,name;

call.student\_details(784, "James", 70, 80, 90);

// Trianglenumber 1

call.triangle1();

// Triangenumber 2

call.triangel2(4);

//Electricity Bill

call.electricity();

//Reverse Array

call.reversearray();

call.copyarray();

call.sumarray();

call.uniques();

call.secondlargest();

call.transpose();

call.prime();

call.countstring();

call.totalcount();

call.fibo(5);

call.armstrong(1);

call.primes(5);

call.stringpalindrome("malayalam");

call.numpalindrome(12521);

}

}