

## ASSESSMENT 1

VASANTHABALAN M

### Exercises

1. Write a C# Sharp program to accept a coordinate point in an XY coordinate system and determine in which quadrant the coordinate point lies.

Input the value for X coordinate :7

Input the value for Y coordinate :9

Expected Output :

The coordinate point (7,9) lies in the First quadrant.

2. Write a C# Sharp program to read roll no, name and marks of three subjects and calculate the total, percentage and division.

Input the Roll Number of the student :784

Input the Name of the Student :James

Input the marks of Physics, Chemistry and Computer Application : 70 80 90

Expected Output :

Roll No : 784

Name of Student : James

Marks in Physics : 70

Marks in Chemistry : 80

Marks in Computer Application : 90

Total Marks = 240

Percentage = 80.00

Division = First

3. Write a program in C# Sharp to calculate and print the Electricity bill of a given customer. The customer id., name and unit consumed by the user should be taken from the keyboard and display the total amount to pay to the customer. The charge are as follow :

Unit Charge/unit

upto 199 @1.20

200 and above but less than 400 @1.50

400 and above but less than 600 @1.80

600 and above @2.00

If bill exceeds Rs. 400 then a surcharge of 15% will be charged and the minimum bill should be of Rs.

100/-

4. Write a program in C# Sharp to make such a pattern like right angle triangle with number increased by 1.

a)

1

2 3

4 5 6

7 8 9 10

b)

1

2 3

4 5 6

7 8 9 10

5. Write a program in C# Sharp to

a. read n number of values in an array and display it in reverse order.

b. copy the elements one array into another array

c. find the sum of all elements of the array

d. count a total number of duplicate elements in an array.

e. print all unique elements in an array

f. find the second largest element in an array

6. Write a program in C# Sharp to find transpose of a given matrix.

7. Write a C# Sharp program that calculates the sum of all prime numbers in an array of numbers

8. Write a program in C# Sharp to count the total number of words in a string.

9. Write a program in C# Sharp to count a total number of alphabets, digits and special characters in a string.

10. Write a C# Sharp program to find the middle character(s) of a given string. Return the middle character if the length of the string is odd and return two middle characters if the length of the string is even.

11. Write a program in C# Sharp to create a function to display the n number Fibonacci sequence.

12. Write a program in C# Sharp to create a function to check if a given number is Armstrong number

or not

13. Write a program in C# Sharp to create a function to check if a given number is Perfect number or not

14. Write a program in C# Sharp to create a function to check if a given number is Prime number or not

15. Write a program in C# Sharp to

a. create a function to check if a given string is Palindrome or not

b. create a function to check if a given number is Palindrome or not

ANSWERS:

MAIN.CS(executable main function)

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;

namespace Assessment1
{
    public class Execute
    {
        public static void Main(String[] args)
        {
            //1.
            //QueSol quadrnt = new QueSol();
            //quadrnt.quad();

            //2.
            //QueSol rnk = new QueSol();
            //rnk.rank();

            //3.
            //QueSol elcty = new QueSol();
            //elcty.electricity();

            //4.a
            //QueSol rgt1 = new QueSol();
            //rgt1.right1();

            //4.b
            //QueSol rgt2 = new QueSol();
            //rgt2.right2();

            //5.a-f
            //QueSol quesA = new QueSol();
            //quesA.QuestA();

            //6
            //QueSol trans = new QueSol();
            //trans.transpose();
        }
    }
}
```

```

//7.
//QueSol prme = new QueSol();
//prme.PrimeSum();

//8.
//QueSol cuntstr = new QueSol();
//cuntstr.CountString();

//9.
//QueSol allcount = new QueSol();
//allcount.AllinOne();

//11.
//QueSol fibo = new QueSol();
//Console.WriteLine("enter the limit:");
//int n = Convert.ToInt32(Console.ReadLine());
//fibo.fiboseries(n);

//12.
//QueSol ms = new QueSol();
//Console.WriteLine("enter the number to check armstrong or not:");
//int num = Convert.ToInt32(Console.ReadLine());
//ms.arms(num);

//14.
//QueSol prm = new QueSol();
//Console.WriteLine("Enter the number to check prime or not:");
//int num = Convert.ToInt32(Console.ReadLine());
//prm.primeNumbers(num);

//15a.
//QueSol strpalind = new QueSol();
//Console.WriteLine("Enter the string to check palindrome or not");
//String word = Console.ReadLine();
//strpalind.strpalindrome(word);

//15b.
QueSol numpali = new QueSol();
Console.WriteLine("Enter the number to check palindrome or not:");
int num = Convert.ToInt32(Console.ReadLine());
numpali.numpalindrome(num);

    }
}

```

QueSol.cs(answer classes)

```

using System;

using System.Collections.Generic;

using System.Globalization;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

```

```

namespace Assessment1
{
    internal class QueSol
    {
        public void quad()
        {
            int xcord, ycord;

            Console.WriteLine("Input the value of X co-ordinate:");
            xcord = Convert.ToInt32(Console.ReadLine());

            Console.WriteLine("Input the value of Y co-ordinate:");
            ycord = Convert.ToInt32(Console.ReadLine());

            if(xcord>0 && ycord > 0)
            {
                Console.WriteLine("First Quadrant");
            }
            else if(xcord<0 && ycord > 0)
            {
                Console.WriteLine("Second Quadrant");
            }
            else if(xcord<0 && ycord < 0)
            {
                Console.WriteLine("Third Quadrant");
            }
            else{
                Console.WriteLine("Fourth Quadrant");
            }

        }

        public void rank() {
            int rollno, subject1, subject2, subject3, total;

```

```

float percentage;

string name;

Console.WriteLine("Enter the rollnumber,name,sub1mark,sub2mark,sub3mark");

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

name = Console.ReadLine();

subject1 = Convert.ToInt32(Console.ReadLine());

subject2 = Convert.ToInt32(Console.ReadLine());

subject3 = Convert.ToInt32(Console.ReadLine());

total = subject1 + subject2 + subject3;

percentage = total / 3;

if (percentage >= 80)
{
    Console.WriteLine("First");
}

else if (percentage >= 75 && percentage<80)
{
    Console.WriteLine("Second");
}

else if (percentage>=40 && percentage < 75)
{
    Console.WriteLine("Third");
}

else
{
    Console.WriteLine("Fourth");
}

}

public void electricity()
{
    Console.WriteLine("Enter the customerid,customer name and unit counsumed");
}

```

```
int customerId = Convert.ToInt32(Console.ReadLine());

string name;

name = Console.ReadLine();

double unitcons = Convert.ToDouble(Console.ReadLine());

if (unitcons <= 199)
{
    unitcons *= 1.20;
    if (unitcons > 400)
    {
        double surcharge = 0.15, newsurcharge=0, charge=0;
        charge = unitcons * surcharge;
        newsurcharge = unitcons + charge;
        Console.WriteLine(newsurcharge);

    }
    else
    {
        Console.WriteLine(unitcons);
    }
}

else if ((unitcons >= 200) && (unitcons<400))
{
    unitcons *= 1.50;
    if (unitcons > 400)
    {
        double surcharge = 0.15, newsurcharge = 0, charge = 0;
        charge = unitcons * surcharge;
        newsurcharge = unitcons + charge;
        Console.WriteLine(newsurcharge);

    }
}
```

```
else
{
    Console.WriteLine(unitcons);
}
}
else if ((unitcons >= 400) && (unitcons < 600))
{
    unitcons *= 1.80;
    if (unitcons > 400)
    {
        double surcharge = 0.15, newsurcharge = 0, charge = 0;
        charge = unitcons * surcharge;
        newsurcharge = unitcons + charge;
        Console.WriteLine(newsurcharge);

    }
    else
    {
        Console.WriteLine(unitcons);
    }
}
else if (unitcons >= 600)
{
    unitcons *= 2.00;
    if (unitcons > 400)
    {
        double surcharge = 0.15, newsurcharge = 0, charge = 0;
        charge = unitcons * surcharge;
        newsurcharge = unitcons + charge;
        Console.WriteLine(newsurcharge);
    }
}
```



```

    }
    else
    {
        Console.WriteLine(unitcons);
    }
}

```

```

}
public void right1()
{
    Console.Write("enter the limit:");
    int num = 1, limit = Convert.ToInt32(Console.ReadLine());
    for(int row=1; row <=limit ; row++)
    {

        for(int col = 1; col <= row; col++)
        {
            Console.Write(num+" ");
            num++;
        }
        Console.WriteLine();
    }
}

```

```

}
public void right2()
{
    Console.Write("enter the limit:");
    int num = 1, limit = Convert.ToInt32(Console.ReadLine()),space;
    for (int row = 1; row <= limit; row++)

```

```

    {
        for (space = 1; space <= limit-row; space++)
            Console.Write(" ");
        for (int col = 1; col <= row; col++)
        {
            Console.Write(num+" ");
            num++;
        }
        Console.WriteLine();
    }

}

public void QuestA()
{

    /* Console.Write("Enter the limit of numbers:");
    int limit = int.Parse(Console.ReadLine());
    int[] arr = new int[limit];
    for (int i = 0; i < limit; i++)
    {
        arr[i] = int.Parse(Console.ReadLine());
    }
    Console.WriteLine(string.Join(", ", arr));
    for (int i = 0, j = arr.Length - 1; i < j; i++, j--)
    {

        int reverse = arr[i];
        arr[i] = arr[j];
        arr[j] = reverse;
    }
}

```

```

    }

    Console.Write(string.Join(", ", arr));*/

//2.

int[] arr1 = new int[] { 1, 2, 3, 4, 5 };
int[] arr2 = new int[arr1.Length];

// Using Array.Copy
Array.Copy(arr1, arr2, arr1.Length);

// Using foreach loop
for (int i = 0; i < arr1.Length; i++)
{
    arr2[i] = arr1[i];
}

/*3.

int[] arr = new int[] { 21, 12, 53, 34, 52 };
int sum = 0;

for (int i = 0; i < arr.Length; i++)
{
    sum += arr[i];
}

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

*/

/*4.

int[] arr = new int[] { 11, 23, 11, 45, 22, 22, 11 };

```

```
Dictionary<int, int> dict = new Dictionary<int, int>();
```

```
foreach (int num in arr)
{
    if (dict.ContainsKey(num))
    {
        dict[num]++;
    }
    else
    {
        dict.Add(num, 1);
    }
}
```

```
foreach (KeyValuePair<int, int> kvp in dict)
{
    if (kvp.Value > 1)
    {
        Console.WriteLine("Number {0} appears {1} times.", kvp.Key, kvp.Value);
    }
}*/
```

```
/*5.
```

```
int[] arr1 = new int[] { 11,23,11,45,22,22,11};
Dictionary<int, int> dict = new Dictionary<int, int>();
```

```
foreach (int num in arr)
{
    if (dict.ContainsKey(num))
    {
        dict[num]++;
    }
}
```

```

    }
    else
    {
        dict.Add(num, 1);
    }
}

foreach (KeyValuePair<int, int> kvp in dict)
{
    if (kvp.Value == 1)
    {
        Console.WriteLine(kvp.Key);
    }
}
*/

/*6.

int[] arr2 = new int[] { 21,4,32,45,67 };
Array.Sort(arr2);
int secondLargest = arr2[arr2.Length - 2];

Console.WriteLine("The second largest element in the array is: {0}", secondLargest);

//Console.WriteLine("Enter the correct choices 1-6 only");

}
*/

}

```

```

public 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] + " ");
        }

        Console.WriteLine();
    }

}

public void PrimeSum()
{
    int[] arr = new int[] { 1, 2, 3, 4, 5 };
    int sum = 0;

    // Function to check if a number is prime
    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;
    }
}

```

```

// Loop through the array and add up the prime numbers
foreach (int num in arr)
{
    if (IsPrime(num))
    {
        sum += num;
    }
}

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

public void CountString()
{
    string str = "This is a sample string with multiple words.";

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

    // Count the number of words
    int wordCount = words.Length;

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

public void AllinOne()
{
    string str = "To !dentify 1234 as digit$";

    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);
```

```
}
```

```
public void fiboseries(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;  
}  
}
```

```
public void arms(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");  
        }  
    }  
}
```

```
    }  
}
```

```
public void primeNumbers(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");  
        }  
    }  
}
```

```
public void strpalindrome(String word)  
{  
    bool isPalindrome = true;  
    int i = 0;  
    int j = word.Length - 1;  
  
    while (i < j)  
    {  
        if (word[i] != word[j])  
        {  
            isPalindrome = false;  
        }  
    }  
}
```

```

        isPalindrome = false;

        break;
    }

    i++;
    j--;
}

if (isPalindrome)
{
    Console.WriteLine("The string is a palindrome.");
}
else
{
    Console.WriteLine("The string is not a palindrome.");
}

}

public 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.");  
}  
}  
  
}  
  
}
```