Final Task:

Exercise 1:

1. Create a C# application calculates the sum of the two provided integer values and returns triple of the sum of the two numbers if they are equal.

```
Console.WriteLine("Enter two numbers: ");
Console.WriteLine("First number: ");
int num1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Second number: ");
int num2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(num1!=num2?num1+num2:(num1*2)*3);
```

```
Enter two numbers:
First number:
3
Second number:
4
7
```

```
Enter two numbers:
First number:
3
Second number:
3
18
```

2. Create a C# program to check a student's eligibility for voting by taking into consideration the student's age to be greater than 18.

```
Console.WriteLine("Enter your age: ");
int age = Convert.ToInt32(Console.ReadLine());
Console.WriteLine(age>18?"Eligible":"Not eligible");
```

```
Enter your age:
10
Not eligible
```

```
Enter your age:
20
Eligible
```

3. Create a C# program that will receive a coordinate point as (x,y) and display the quadrant it is in.

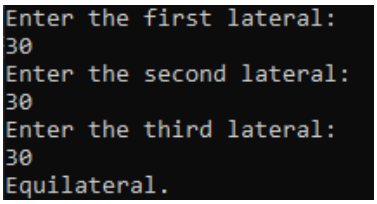
```
Console.WriteLine("Enter the X coordinate: ");
int xCoordinate = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter the Y coordinate: ");
int yCooridnate = Convert.ToInt32(Console.ReadLine());
if (xCoordinate == 0 && yCooridnate == 0) Console.WriteLine("These
coordinates are in the center.");
else if (xCoordinate == 0) Console.WriteLine("These coordinates are on the Y
axis");
else if (yCooridnate == 0) Console.WriteLine("These coordinates are on the X
axis");
else if (xCoordinate > 0 && yCooridnate > 0) Console.WriteLine("First
Quadrant.");
else if (xCoordinate < 0 && yCooridnate > 0) Console.WriteLine("Second
Quadrant.");
else if (xCoordinate < 0 && yCooridnate < 0) Console.WriteLine("Third
Quadrant.");
else if (xCoordinate > 0 && yCooridnate < 0) Console.WriteLine("Fourth
Quadrant.");
```

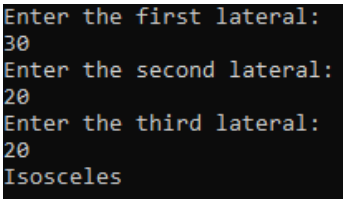
```
Enter the X coordinate:
3
Enter the Y coordinate:
-2
Fourth Quadrant.
```

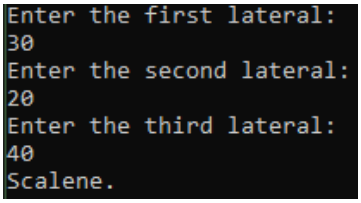
```
Enter the X coordinate:
-3
Enter the Y coordinate:
-2
Third Quadrant.
```

4. Write a C# program asks user to input the laterals **only** for triangle and then display the type of triangle if it is right, isosceles, or equilateral.

```
Console.WriteLine("Enter the first lateral: ");
int lat1 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter the second lateral: ");
int lat2 = Convert.ToInt32(Console.ReadLine());
Console.WriteLine("Enter the third lateral: ");
int lat3 = Convert.ToInt32(Console.ReadLine());
if (lat1 == lat2 && lat2 == lat3 && lat1 == lat3)
Console.WriteLine("Equilateral.");
else if (lat1 != lat2 && lat2 != lat3 && lat1 != lat3)
Console.WriteLine("Scalene.");
else Console.WriteLine("Isosceles");
```

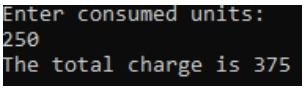


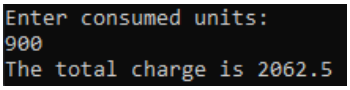


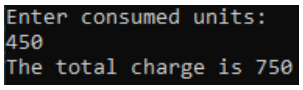


5. Create a C# program to compute and print customer's electricity bill. The entire amount to be paid to is calculated based on units consumed by the client which will be extracted from keyboard. The fees are as shown in table below and note that 10% will be added as surcharge if units consumed exceed 600 watts.

```
Console.WriteLine("Enter consumed units: ");
int units = Convert.ToInt32(Console.ReadLine());
int ogUnits = units;
double charge = 0;

if (units > 450)
{
    charge += ((units-450)*2.5);
    units = 450;
}

if (units > 300)
{
    charge += ((units-300) * 2);
    units = 300;
}

if (units <= 300)
{
    charge += (units * 1.5);
}

if (ogUnits > 600)
{
    charge += (charge * 0.1);
}

Console.WriteLine($"The total charge is {charge}");
```







Exercise 2:

Create a Bank System for ATM enables clients to Deposit, Withdraw and Check their balance considering all validations may occur on user inputs such as (valid value, greater than zero and less than balance, etc.…) .

Note: Make sure the menu will appear on screen till the client enters exit or similar.

```
double balance = 5000;
Console.WriteLine("Welcome to the bank system: ");
while (true)
{
    Console.WriteLine("Please select an option:");
    Console.WriteLine("1. Check Balance");
    Console.WriteLine("2. Deposit");
    Console.WriteLine("3. Withdraw");
    Console.WriteLine("4. Exit");
    int input = Convert.ToInt32(Console.ReadLine());
    if (input == 4)
    {
        Console.WriteLine("See you later!");
        break;
    }
    else if (input == 1)
    {
        Console.WriteLine("Your current balance is " + balance);
    }
    else if (input == 2)
    {
        Console.WriteLine("Enter the amount of money you want to deposit: ");
        double deposit = Convert.ToDouble(Console.ReadLine());
        if (deposit <= 0)
        {
            Console.WriteLine("Invalid amount, please try again.");
        }
        else
        {
```

```
            balance += deposit;
            Console.WriteLine("Transaction successfull.");
            Console.WriteLine($"Available balance is: {balance}");
        }
    }
    else if (input == 3)
    {
        Console.WriteLine("Enter the amount of money you want to withdraw: ");
        double withdraw = Convert.ToDouble(Console.ReadLine());
        if (withdraw <= 0 || withdraw > balance)
        {
            Console.WriteLine("Invalid Transaction, please try again later.");
        }
        else
        {
            balance-=withdraw;
            Console.WriteLine("Transaction successfull.");
            Console.WriteLine($"Available balance is: {balance}");
        }

    }
}
```

```
Welcome to the bank system:
Please select an option:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
1
Your current balance is 5000
Please select an option:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
3
Enter the amount of money you want to withdraw:
6000
Invalid Transaction, please try again later.
Please select an option:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
3
Enter the amount of money you want to withdraw:
2000
Transaction successfull.
Available balance is: 3000
Please select an option:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
2
Enter the amount of money you want to deposit:
7000
Transaction successfull.
Available balance is: 10000
Please select an option:
1. Check Balance
2. Deposit
3. Withdraw
4. Exit
4
See you later!
```

Exercise 3:

Create a program for taking the number of students dynamically at a class with their grades from the teacher and then show the minimum mark, maximum mark and the average one. Plus, how many students are above the Average and how much below Average.

```
List<int> grades = new List<int>();
int grade = 0, sum = 0;
Console.WriteLine("Please enter '-1' when you have entered the grades of all
students.");
while (grade != -1)
{
    grade = Convert.ToInt32(Console.ReadLine());
    if (grade == -1) break;
    grades.Add(grade);
    sum+= grade;
}
grades.Sort();
Console.WriteLine($"The maximum grade is {grades[grades.Count-1]}");
Console.WriteLine($"The minimum grade is {grades[0]}");
Console.WriteLine($"The average grade is {sum/grades.Count}");
int gt = 0, lt = 0;

foreach (int g in grades)
{
    if (g > sum / grades.Count)
        gt++;
    else if (g < sum / grades.Count)
        lt++;
}

Console.WriteLine($"Number of grades greater than the average: {gt}");
Console.WriteLine($"Number of grades less than the average: {lt}");
```

```
Please enter '-1' when you have entered the grades of all students.
75
85
90
88
70
92
65
80
95
78
-1
The maximum grade is 95
The minimum grade is 65
The average grade is 81
Number of grades greater than the average: 5
Number of grades less than the average: 5
```

Jagged array:

```
int[][] arr = new int[3][];

arr[0] = new int[] { 1, 2, 3 };
arr[1] = new int[] { 4, 5 };
arr[2] = new int[] { 6, 7, 8, 9 };

for (int i = 0; i < arr.Length; i++)
{
    Console.Write($"Row {i} : ");
    for (int j = 0; j < arr[i].Length; j++)
    {
        Console.Write(arr[i][j] + " ");
    }
    Console.WriteLine();
}
```

```
Row 0 : 1 2 3
Row 1 : 4 5
Row 2 : 6 7 8 9
```

Fibonacci Sequence:

```
List<int> fibonacci = new List<int>();
Console.WriteLine("Enter the number of digits in the Fibonacci sequence: ");
int digits = Convert.ToInt32(Console.ReadLine());

fibonacci.Add(0);
fibonacci.Add(1);

for (int i = 2; i < digits; i++)
    fibonacci.Add(fibonacci[i - 1] + fibonacci[i - 2]);


foreach (int i in fibonacci)
    Console.WriteLine(i);
```

```
Enter the number of digits in the Fibonacci sequence:
13
0
1
1
2
3
5
8
13
21
34
55
89
144
```

Throwing an exception made by me:

```
class CustomException : Exception
{
    public CustomException(string message) : base(message) { }
}

class Program
{
    static void Main()
    {
        try
        {
            Console.WriteLine("Enter a number:");
            int num = Convert.ToInt32(Console.ReadLine());

            if (num < 0)
                throw new CustomException("Negative number not allowed!");

            Console.WriteLine("The number entered is " + num);
        }
        catch (CustomException ex)
        {
            Console.WriteLine($"Custom Exception: {ex.Message}");
        }
        catch (FormatException)
        {
            Console.WriteLine("Please enter a valid number.");
        }
    }
}
```

```
Enter a number:
-3
Custom Exception: Negative number not allowed!
```