1. Write a C# Sharp program to print Hello and your name in a separate line.

```
Expected Output :
Hello: Alexandra Abramov
```

2. Write a C# Sharp program to print the sum of two numbers.

3. Write a C# Sharp program to print the result of dividing two numbers.

4. Write a C# Sharp program to print the result of the specified operations. Test data:

```
-1 + 4 * 6
( 35+ 5 ) % 7
14 + -4 * 6 / 11
2 + 15 / 6 * 1 - 7 % 2

Expected Output:
23
5
12
3
```

5. Write a C# Sharp program to swap two numbers.

```
Test Data:
Input the First Number : 5
Input the Second Number : 6
Expected Output:
After Swapping :
First Number : 6
Second Number : 5
```

6. Write a C# Sharp program to print the output of multiplication of three numbers which will be entered by the user.

```
Test Data:
Input the first number to multiply: 2
Input the second number to multiply: 3
Input the third number to multiply: 6
Expected Output:
2 x 3 x 6 = 36
```

7. Write a C# Sharp program to print on screen the output of adding, subtracting, multiplying and dividing of two numbers which will be entered by the user.

```
Test Data:
Input the first number: 25
Input the second number: 4
Expected Output:
25 + 4 = 29
25 - 4 = 21
25 x 4 = 100
25 / 4 = 6
25 mod 4 = 1
```

8. Write a C# Sharp program that takes a number as input and print its multiplication table.

```
Test Data:
Enter the number: 5
Expected Output:
5 * 0 = 0
5 * 1 = 5
5 * 2 = 10
5 * 3 = 15
....
5 * 10 = 50
```

10. Write a C# Sharp program that takes four numbers as input to calculate and print the average.

```
Test Data:
Enter the First number: 10
Enter the Second number: 15
Enter the third number: 20
Enter the four number: 30

Expected Output:
The average of 10 , 15 , 20 , 30 is: 18
```

11. Write a C# Sharp program to that takes three numbers(x,y,z) as input and print the output of (x+y).z and x.y + y.z.

```
Test Data:
Enter first number - 5
Enter second number - 6
Enter third number - 7

Expected Output:
Result of specified numbers 5, 6 and 7, (x+y).z is 77 and x.y + y.z is 72
```

OOP

PhotoAlbum

- Create a class "PhotoAlbum" with a private attribute "numberOfPages."
- It should also have a public method "GetNumberOfPages", which will return the number of pages.
- The default constructor will create an album with 16 pages. There will be an additional constructor, with which we can specify the number of pages we want in the album.
- Create a class "BigPhotoAlbum" whose constructor will create an album with 64 pages.
- Create a test class "AlbumTest" to create an album with its default constructor, one with 24 pages, a "BigPhotoAlbum" and show the number of pages that the three albums have.

Classes Student + Teacher

- Create a new project, and include in it the class Person that you just created.
- Create a class "Student" and another class "Teacher", both descendants of "Person".
- The class "Student" will have a public method "GoToClasses", which will write on screen "I'm going to class."
- The class "Teacher" will have a public method "Explain", which will show on screen "Explanation begins". Also, it will have a private attribute "subject", a string.
- The class Person must have a method "SetAge (int n)" which will indicate the value of their age (eg, 20 years old).
- The student will have a public method "ShowAge" which will write on the screen "My age is: 20 years old" (or the corresponding number).
- You must create another test class called "StudentAndTeacherTest" that will contain "Main" and:

    1. Create a Person and make it say hello
    2. Create a student, set his age to 21, tell him to Greet and display his age
    3. Create a teacher, 30 years old, ask him to say hello and then explain.

Write C# program to

Define interface that 2 method:

```
AccountInfo() //return info;
Process() //process data;
```

Implements the interface just defined only need to mock the data no need to implement the code

Then call in the Main()

Write C# program to

This program mock the use case when connect to the database then user do some transaction. Define interface that method:

```
StartTransaction()
```

Then implement the interface with the class then call in the Main()