Exercice 1 :

Objectif : Ecrire un programme C# servant à calculer la valeur absolue d'un nombre réel x à partir de la définition de la valeur absolue. La valeur absolue du nombre réel x est le nombre réel |x| :

|x| = x , si x ≥ 0

|x| = -x si x < 0

Et si un utilisateur tape une chaine de caractère au lieu d’un nombre ?

Modifiez le code pour l’adapter à ce qu’on a appris aujourd’hui.

Exercice 2

Complétez le code suivant en créant une fonction qui prend deux nombres comme arguments et retourne leur somme.

**Exemple:**

somme(1, 2) ➞ 3

somme(100, 200) ➞ 300

somme(-3, -2) ➞ -5

Écrivez une fonction qui prend la base et la hauteur d’un triangle et retourne sa surface.

Notez que la surface d’un triangle est: (base \* hauteur) / 2

Exemple:  
getSurface(8, 2) ➞ 8

getSurface(7, 3) ➞ 10

using System;

namespace Triangle

{

public class Program

{

public static int getSurface(int b, int h)

{

// Écrivez votre code ici

}

// Testez le code

public static void Main()

{

Console.WriteLine(getSurface(8, 2));

Console.WriteLine(getSurface(7, 3));

}

}

}

Exercice 4:

Écrivez un programme C# pour renvoyer le reste de deux nombres. Il existe un seul opérateur en C#, capable de fournir le reste d’une division. Deux nombres sont transmis comme paramètres. Le premier paramètre divisé par le deuxième paramètre.

Exemple:  
resteDiv(1, 3) ➞ 1

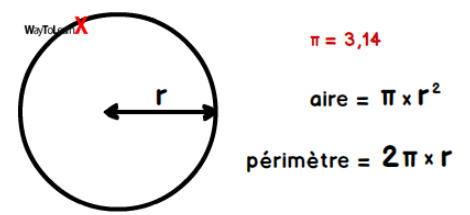
resteDiv(2, 4) ➞ 2

resteDiv(3, 3) ➞ 0

Exercice 5 :  
  
1- Ecrire un programme en c qui permet d'afficher les jours de la semaine (lundi, mardi,...) sachant que les jours sont codés de 1 à 7:  
1----------------->Lundi  
2----------------->Mardi  
......  
7---------------->Dimanche

Exercice 6 :

Votre tâche consiste à créer un constructeur Cercle qui crée un cercle avec un rayon fourni par un argument.   
Les cercles construits doivent avoir deux getters GetArea() (PIr ^ 2) et GetPerimeter() (2PI \* r) qui donnent à la fois l’aire et le périmètre respectifs.



Exercice 7 :

– Créez une classe « Person »  
– Créez une classe « Student » et une autre classe « Teacher », les deux héritent de la classe « Person ».  
– La classe « Student » aura une méthode publique « GoToClasses », qui affichera à l’écran « Je vais aller à ma classe. ».  
– La classe « Teacher » aura une méthode publique « Explain », qui affichera à l’écran «Début de l’explication». En plus, il aura un attribut privé « subject » de type string.  
– La classe « Person » doit avoir une méthode « SetAge(int n) » qui indiquera la valeur de leur âge (par exemple, 15 ans).  
– La classe « Student » aura une méthode publique « DisplayAge » qui écrira sur l’écran « Mon age est : XX ans ».  
– Vous devez créer une autre classe de test appelée « Test » qui contiendra « Main » et:  
– Créez un objet Person et faites-lui dire « Allo ! »  
– Créer un objet Student, définir son âge à 15 ans, faites-lui dire « Allo ! », « Je vais aller à ma classe. » et afficher son âge  
– Créez un objet Teacher, 40 ans, demandez-lui de dire « Allo ! » puis commence l’explication.