

6.1)

N.B. Votre fonction `accumulator()` peut être appelée dans une fonction nommée `E01()`

Solution 01

```
public static float Accumulator(float num, float total)
{
    total += num;
    return total;
}
0 references
public static void E01()
{
    float total = 0.0f;
    for (int i = 1; i <= 5; i++)
    {
        Console.WriteLine($"Entrer le nombre {i} : ");
        total = Accumulator(float.Parse(Console.ReadLine()), total);
    }
    Console.WriteLine($"La somme est {total}");
}

static void Main(string[] args)
{
    E01(); //l'appel de la fonction
```

Solution 2

```
0 references
public static float Accumulator(float num1 = 0f, float num2 = 0f, float num3 = 0f, float num4 = 0f, float num5 = 0f)
{
    return (num1 + num2 + num3 + num4 + num5);
}
3 references
public static void E01()
{
    float total = 0.0f;
    Console.WriteLine($"Entrer les nombres : ");
    total = Accumulator(float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()), float.Parse(Console.ReadLine()));
    Console.WriteLine($"La somme est {total}");
}
0 references
static void Main(string[] args)
{
    E01(); //l'appel de la fonction
```

6.2)

N.B. Votre fonction peut être appelée dans une fonction nommée E02()

```
1 reference
private static float Sum(float[] tableau) {
    float s = 0;
    for (int i = 0; i < tableau.Length; i++) { s += tableau[i]; }
    return s;
}
1 reference
public static void E02(int tailleTableau=5)
{
    float[] tableau = new float[tailleTableau];
    for (int i = 0; i < tableau.Length; i++)
    {
        Console.Write($"Entrer nombre #{i + 1} : ");
        tableau[i] = float.Parse(Console.ReadLine());
    }
    Console.WriteLine($"La somme de ces {tailleTableau} nombres est de {Sum(tableau)}");
}

0 references
static void Main(string[] args)
{
    |
    |
    |
    E02();//l'appel de la fonction
}
```

6.3)

```
private static double Distance(double x1, double y1, double x2, double y2)
{
    return Math.Sqrt(Math.Pow((y2 - y1),2) + Math.Pow((x2 - x1),2));
}
1 reference
public static void E03()
{
    int x1, x2, y1, y2;
    Console.Write($"Entrez x1: ");
    x1 = int.Parse(Console.ReadLine());
    Console.Write($"Entrez y1: ");
    y1 = int.Parse(Console.ReadLine());
    Console.Write($"Entrez x2: ");
    x2 = int.Parse(Console.ReadLine());
    Console.Write($"Entrez y2: ");
    y2 = int.Parse(Console.ReadLine());

    Console.WriteLine($"La distance entre les points est {Math.Round(Distance(x1, y1, x2, y2), 2)}");
}
0 references
static void Main(string[] args)
{
    E03();//l'appel de la fonction
}
```

6.4)

```
private static int chiffresAleatoire()
{
    Random rndNum = new Random();
    return rndNum.Next(0, 100);
}

1 reference
public static int[] RandomNumbers(int nombreDeChiffres)
{
    try
    {
        int[] tableau = new int[nombreDeChiffres];

        for (int i = 0; i < tableau.Length; i++)
        {
            tableau[i] = chiffresAleatoire();
        }
        return tableau;
    }
    catch
    {
        int[] tableau = new int[0];
        return tableau;
    }
}

1 reference
```

```
public static void E04()
{
    int quantite = 0;
    Console.WriteLine($"Entrer une quantité de nombre : ");
    try
    {
        quantite = int.Parse(Console.ReadLine());
    }
    catch
    {
        quantite = chiffresAleatoire();
    }

    int[] tableau = RandomNumbers(quantite);
    Console.WriteLine("Mes nombres sont : ");
    for (int i = 0; i < tableau.Length; i++)
    {
        Console.WriteLine($"{tableau[i]} ");
    }
}

References
static void Main(string[] args)
{
    E04(); //l'appel de la fonction
}
```

6.5)

N.B. Votre fonction peut être appelée dans une fonction nommée E05()

```
private static float Power(float nombre, int power) {
    float P = 1; ;
    for(int i = 0; i < power; i++) P *= nombre;
    return P;
}
1 reference
public static void E05()
{
    float nombre;
    int puissance;
    char choix = '0';

    do
    {
        Console.WriteLine("Entrer un calcul (O/N)? ");
        do
        {
            choix = Console.ReadLine()[0];
        } while (choix != 'o' || choix != 'O' || choix != 'N' || choix != 'n');
        if (choix == 'o' || choix == 'O')
        {
            Console.Write("Entrer nombre : ");
            nombre = float.Parse(Console.ReadLine());
            Console.Write("Entrer puissance : ");
            puissance = int.Parse(Console.ReadLine());
            Console.WriteLine($"Résultat = {Power(nombre, puissance)}");
        }
        else if (choix == 'n' || choix == 'N') return;
    } while (choix != 'o' || choix != 'O');
}
0 references
static void Main(string[] args)
{
    E05();//l'appel de la fonction
}
```

6.6)

```
private static float Power(float nombre, int power) {
    float P = 1; ;
    for(int i = 0; i < power; i++) P *= nombre;
    return P;
}
//reference
public static void E05()
{
    float nombre;
    int puissance;
    bool choix;

    do
    {
        do
        {
            choix = AnotherPower();
        } while (!choix);
        if (choix)
        {
            Console.Write("Entrer nombre : ");
            nombre = float.Parse(Console.ReadLine());
            Console.Write("Entrer puissance : ");
            puissance = int.Parse(Console.ReadLine());
            Console.WriteLine($"Résultat = {Power(nombre, puissance)}");
        }
        else if (!choix) return;
    } while (!choix);
}
//reference
static public bool AnotherPower()
{
    char choix;
    Console.WriteLine("Entrer un calcul (O/N)? ");
    choix = Console.ReadLine()[0];
    if (choix == 'N' || choix == 'n') return false;
    else if (choix == 'o' || choix == 'O') return true;
    else return false;
}

//References
static void Main(string[] args)
{
    E05(); //l'appel de la fonction
}
```

Les fonctions avancées

6.7)

```
static public int factorielle(int nombre)//fonction réursive est une fonction qui appelle elle même
{
    if (nombre > 1) return nombre * factorielle(nombre - 1);
    else return 1;
}
```

1 reference

```
public static void E07()
```

```
{
    int nombre;

    Console.Write("Entrer un nombre entier : ");
    nombre = int.Parse(Console.ReadLine());

    int resultat = factorielle(nombre);

    Console.WriteLine($"La factorielle de {nombre} est {resultat}");
}
```

0 references

```
static void Main(string[] args)
```

```
{
    E07();//l'appel de la fonction
}
```

6.8)

```
1 reference
static public bool Prime(int chiffre)
{
    int possibilite = 2;
    for (int i = 2; i < chiffre; i++)
    {
        if (chiffre % possibilite == 0)
        {
            return false;
        }
    }
    return true;
}
1 reference
public static void E08()
{
    int entree = 1;
    while (entree != 0)
    {
        Console.Write("Entrer nombre : ");
        entree = int.Parse(Console.ReadLine());
        bool resultat = Prime(entree);
        if (entree == 0) break;
        else if (resultat) Console.WriteLine($"{entree} est un nombre premier");
        else Console.WriteLine($"{entree} est un nombre divisible");
    }
}
References
static void Main(string[] args)
{
    E08();//l'appel de la fonction
    //Affichage du resultat de la fonction
}
```

6.9)


```
static public double Fibonacci(int chiffre)
{
    /* calcul de fibonnacci */
    if (chiffre == 0) return 0;
    else if (chiffre == 1 || chiffre == 2) return 1;
    else return (Fibonacci(chiffre - 1) + Fibonacci(chiffre - 2));
}
1 reference
public static void E09()
{
    int position;
    Console.Write("Entrer la position de Fibonacci : ");
    position = int.Parse(Console.ReadLine());
    Console.WriteLine($"Nombre #{position} de la suite Fibonacci = {Fibonacci(position)}");
}
0 references
static void Main(string[] args)
{
    E09();//l'appel de la fonction
```

6.10)

À votre imagination...