

6.1)

```
0 references
public static bool Check(string str)
{
    return str.EndsWith("s");
}

static void Main(string[] args)
{
    if (Check("chaines")) Console.WriteLine("pluriel");
    else Console.WriteLine("singulier...");
    //hute cmn = 0.
}
```

6.2)

```
1 reference
public static int getSmall(int[] tab)
{
    int min=tab[0];
    for(int i=1; i<tab.Length; i++)
    {
        //Console.WriteLine("AVANT MIN: {0} et {1} ", min, tab[i]);
        min = Math.Min(tab[i], min);
        //Console.WriteLine("APRES MIN: {0} ", min);
    }
    return min;
}

0 references
static void Main(string[] args)
{
    int[] tab = new int[]{1,4,-6,10,7,-8};
    Console.WriteLine("le min du tableau : "+getSmall(tab));
    //if (Check("le min du tableau : "+getSmall(tab))) Console.WriteLine("pluriel");
}
```

6.3)

```
1 reference
public static int getBigNbr(int[] tab)
{
    int max = tab[0];
    for (int i = 1; i < tab.Length; i++)
    {
        //Console.WriteLine("AVANT MIN: {0} et {1} ", min, tab[i]);
        max = Math.Max(tab[i], max);
        //Console.WriteLine("APRES MIN: {0} ", min);
    }
    return max;
}
0 references
static void Main(string[] args)
{
    int[] tab = new int[] {1,4,-6,10,7,-8};
    Console.WriteLine("le max du tableau : "+getBigNbr(tab));
}
```

6.4)

```
public static double Carre(double N) { return N * N; }
1 reference
public static double Cube(double N) { return N * Carre(N); /*ou bien return Math.Pow(N,3);*/ }

0 references
static void Main(string[] args)
{
    do
    {
        Console.Write("Entrer un nombre :");
        double nbr = double.Parse(Console.ReadLine());
        Console.WriteLine("le carre de {0} est : {1}", nbr, Carre(nbr));
        Console.WriteLine("le cube de {0} est : {1}", nbr, Cube(nbr));
    } while (true);
    //int[] tab = new int[] {1,4,-6,10,7,-8};
}
```

6.5)

```
public static bool estEgal(int[] tab1, int[] tab2)
{
    if (tab1.Length == tab2.Length)
    {
        for (int i = 0; i < tab2.Length; i++)
        {
            if (tab1[i] != tab2[i]) return false;
        }
    }
    else return false;
    return true;
}

References
static void Main(string[] args)
{
    int[] tab1 = new int[] { 1, 3, 5, 6, 7 };
    int[] tab2 = new int[] { 1, 3, 5, 6, 8 };
    if (estEgal(tab1, tab2)) Console.WriteLine("les tableaux sont égaux.");
    else Console.WriteLine("les tableaux ne sont pas égaux.");
}
```

6.6)

```
public static bool RechercheChar(char ARechercher, char[] tabChars)
{
    for (int i = 0; i < tabChars.Length; i++) if (tabChars[i] == ARechercher) return true;
    return false;
}

References
static void Main(string[] args)
{
    char[] tab1 = new char[] { 'a', 'z', 'h', 'n', 'd' };
    if (RechercheChar('n', tab1)) Console.WriteLine("la lettre existe dans le tableau.");
    else Console.WriteLine("la lettre n'existe pas dans le tableau.");
}
```

6.7)

```
public static int RechercheCharOccurence(char ARechercher, char[] tabChars)
{
    int compteur = 0;
    for (int i = 0; i < tabChars.Length; i++) if (tabChars[i] == ARechercher) compteur++;
    return compteur;
}

References
static void Main(string[] args)
{
    char[] tab1 = new char[] { 'a', 'z', 'n', 'n', 'd' };
    if (RechercheCharOccurence('n', tab1) != 0) Console.WriteLine("la lettre existe dans le tableau {0} fois.", RechercheCharOccurence('n', tab1));
    else Console.WriteLine("la lettre n'existe pas dans le tableau.");
}
```