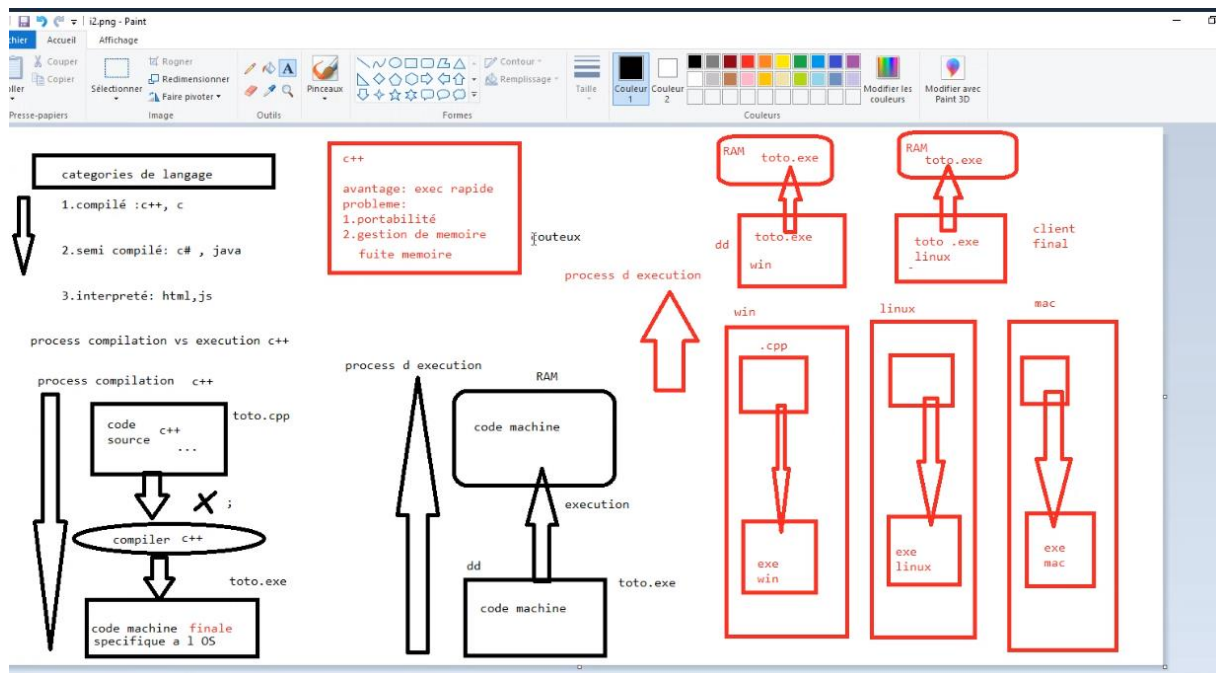
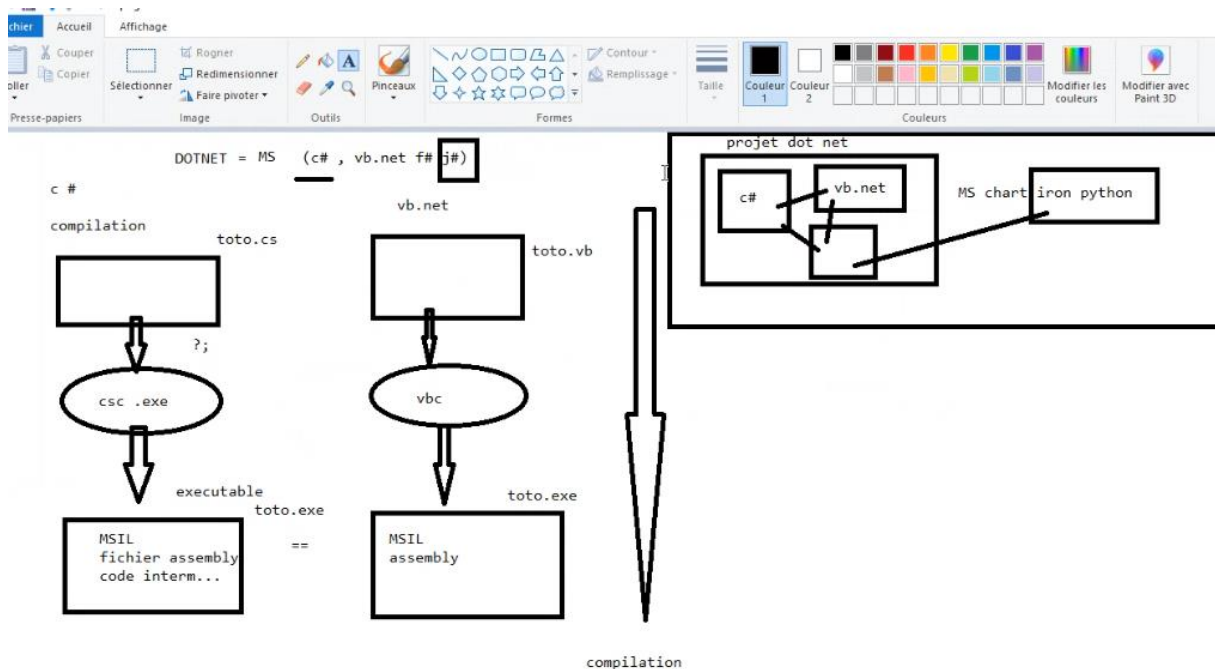
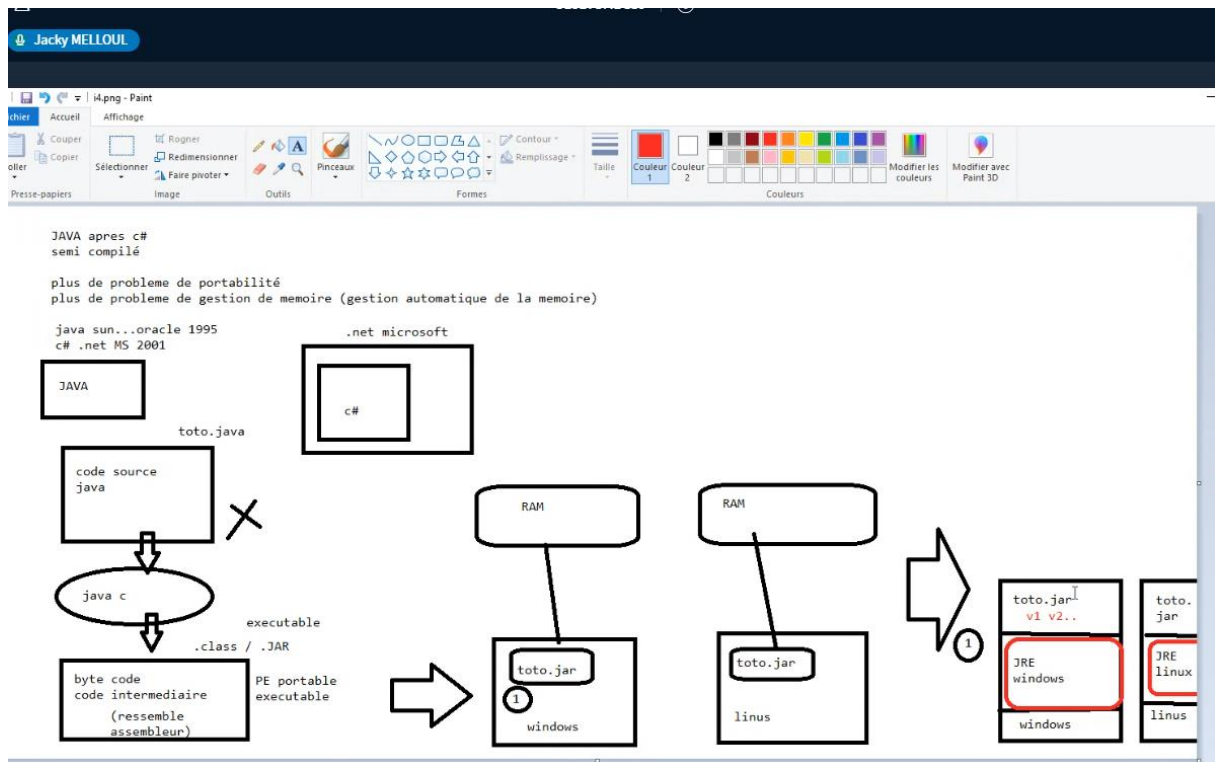


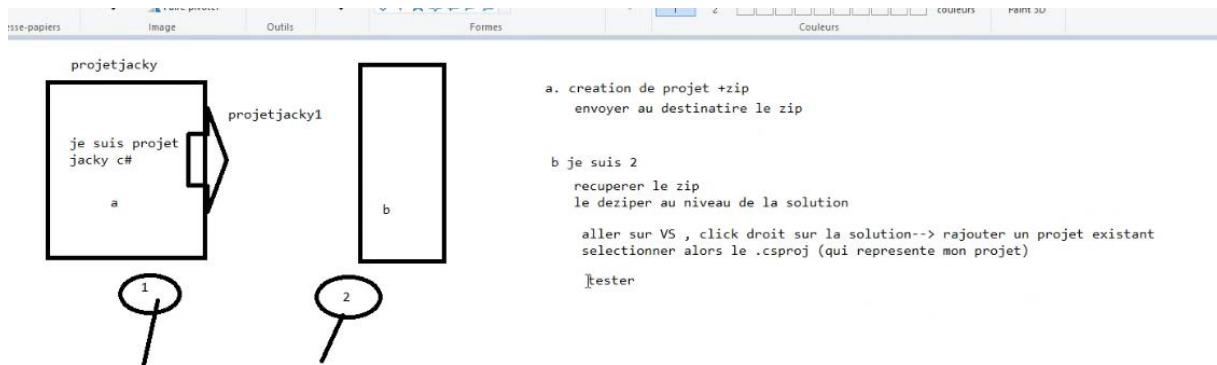
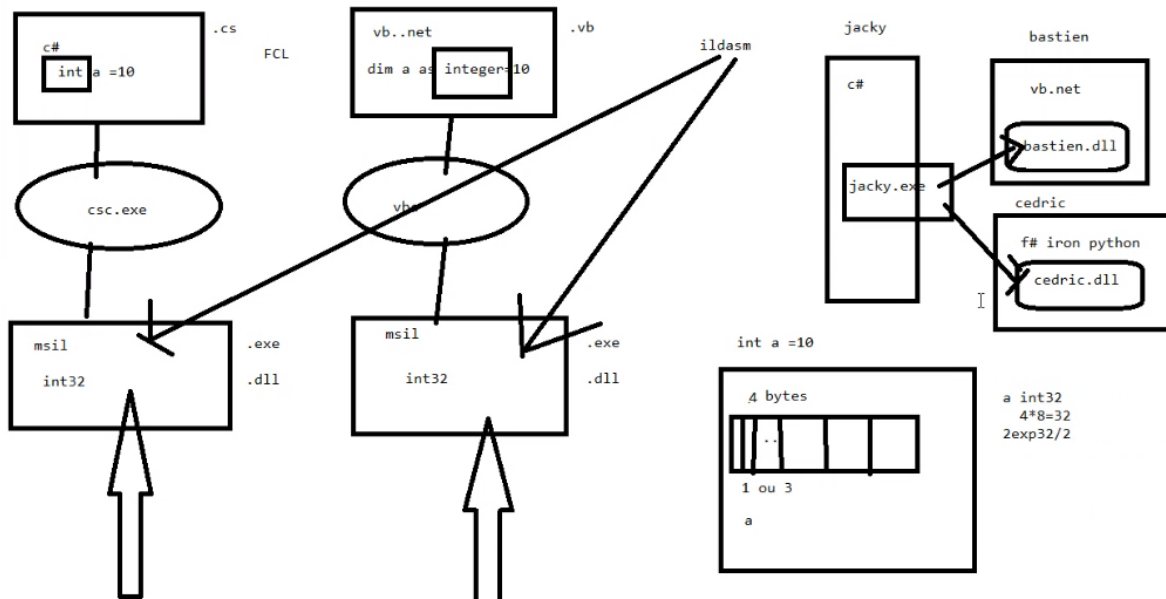
Developpement .net

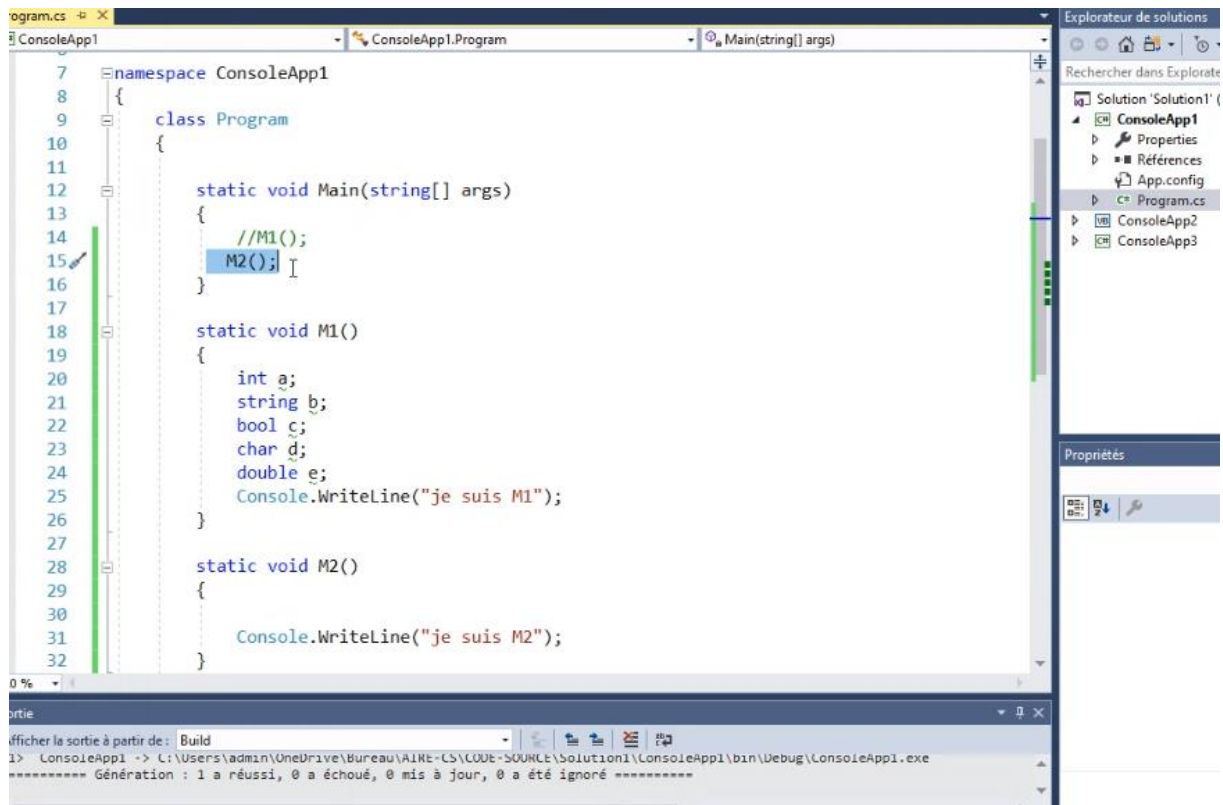


DEBUT	FIN	MODULE	EVALUATION	CODE
10/10/2023	11/10/2023	Rôle et comportement du consultant objet	11/10/2023	6017
12/10/2023	13/10/2023	Modélisation normalisée	13/10/2023	5734
16/10/2023	20/10/2023	Big Data	20/10/2023	3463
23/10/2023	27/10/2023	SERVER SQL : Transact SQL et Exploitation	27/10/2023	5233
30/10/2023	30/10/2023	Travail en équipe	30/10/2023	4903
31/10/2023	03/11/2023	Algorithmie	03/11/2023	4529
06/11/2023	08/11/2023	Objet UML	08/11/2023	4762
09/11/2023	15/11/2023	Le développement .NET EN C# sous VISUAL	15/11/2023	6296
16/11/2023	22/11/2023	Conception d'interfaces graphiques AVEC	22/11/2023	4465
23/11/2023	27/11/2023	Design Patterns	27/11/2023	3037
28/11/2023	28/11/2023	Gestion du temps et des priorités	28/11/2023	7779
29/11/2023	29/11/2023	Conduite de réunion	29/11/2023	5324
30/11/2023	30/11/2023	Présenter ses nouvelles compétences	30/11/2023	3851
01/12/2023	07/12/2023	L'accès aux données C# sous VISUAL STL	07/12/2023	7975
08/12/2023	14/12/2023	Projet C# IHM BDD	14/12/2023	6967
15/12/2023	21/12/2023	Initiation Web avec HTML5, CSS3, JAVASC	21/12/2023	9540
22/12/2023	26/12/2023	XML	26/12/2023	6581
27/12/2023	03/01/2024	Conception d'interfaces graphiques Web	03/01/2024	5355
04/01/2024	05/01/2024	Introduction à ASP.NET Core 2.1	05/01/2024	6732
08/01/2024	12/01/2024	Projet ASP.NET et Services Web	12/01/2024	6806
15/01/2024	16/01/2024	Web Services REST avec .NET	16/01/2024	2718
17/01/2024	18/01/2024	Développer des applications Windows Azu	18/01/2024	2673
19/01/2024	25/01/2024	Angular	25/01/2024	8694
26/01/2024	01/02/2024	Projet Final & Soutenance ? C#	01/02/2024	5524

Les informations figurant sur ce document sont la propriété exclusive d'AJC Formation. Étant réputées confidentielles, leur consultation est réservée exclusivement aux personnes autorisées par AJC Formation. Aucune copie, aucun usage, aucune







- using System;
- using System.Collections.Generic;
- using System.Linq;
- using System.Text;
- using System.Threading.Tasks;
-
- namespace ConsoleApp1
- {
- class Program
- {
-
- static void Main(string[] args)
- {
- M5();
- }
- static void M5()
- {
- string path1 = @"C:\tmp\test.txt";
- string path2 = "C:\\tmp\\test.txt";
-
- Console.WriteLine(path1);
- Console.WriteLine(path2);
-
- }
-
- static void M4()

```

{
    string str1 = "bonjour ";
    string str2 = "toto";
    string str3 = str1 + "\t"+str2;
    Console.WriteLine(str3);
}

static void M3()
{
    int a = 1000000;
    string b = "jhgsdhgzhcdghz";
    bool c = false;
    double d = 10.5;

    Console.WriteLine(a);
    Console.WriteLine(b);
    Console.WriteLine(c);
    Console.WriteLine(d);
}

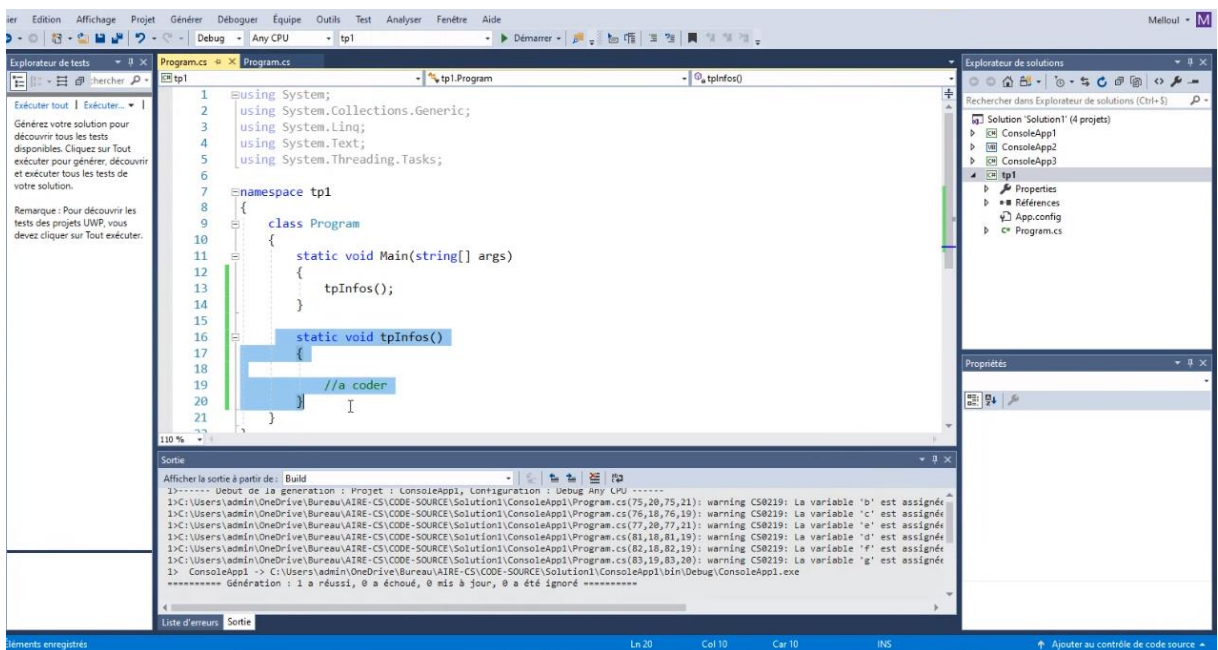
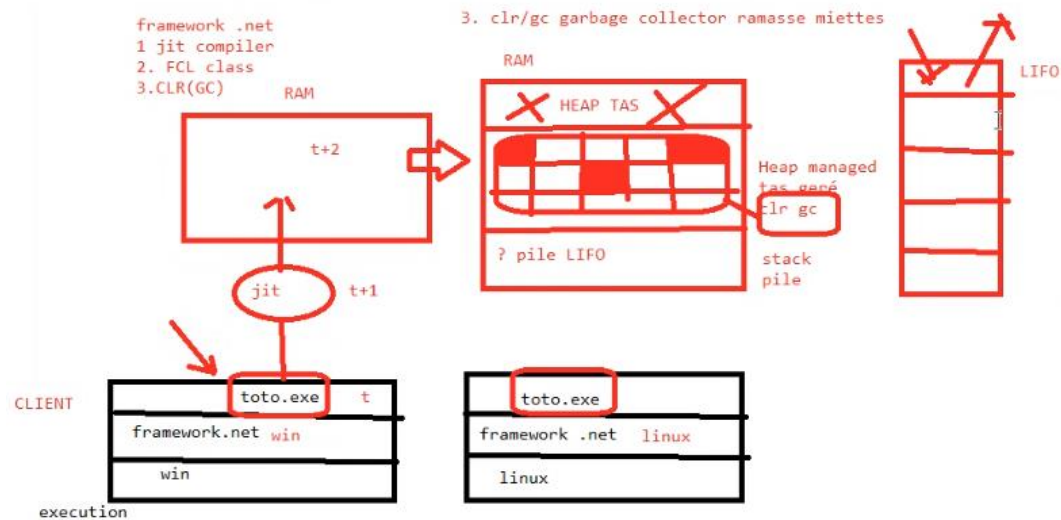
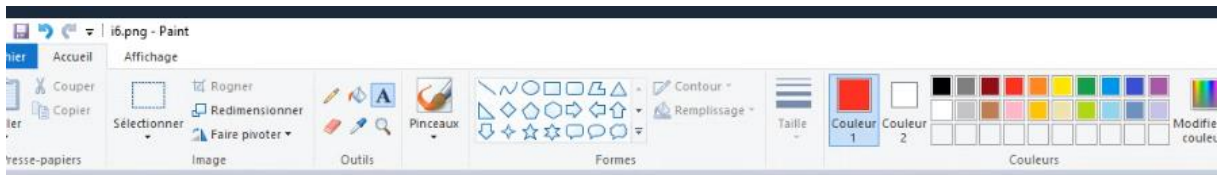
static void M2()
{
    string str1 = "bonjour \t\t toto";
    string str2 = "bonjour \n titi";
    Console.WriteLine(str1);
    Console.WriteLine("-----");
    Console.WriteLine(str2);
}

static void M1()
{
    int a=1000000;
    string b="jhgsdhgzhcdghz";
    bool c = false;
    double e=10.5;
    Console.WriteLine("je suis M1");
    Console.WriteLine(a);

    char d = 'a';
    long f = 10;// int64 2exp 64 /2
    short g = 10; //16
}

```

-
- }
- }



il

TpInfos

n



dans le projet tp:

creer une methode TpInfos (qui sera appelé dans le main)

qui demande a l'utilisateur de saisir son nom et son prenom

faire une concatenation du nom et prenom

afficher en reponse finale :|HELLO et nom prenom avec tabulation

livrer uniquement un fichier tpInfos.txt

```
//M6();
TpInfo();

static void TpInfo()

{
    Console.WriteLine("Merci de saisir votre nom : ");
    string nom = Console.ReadLine();
    Console.WriteLine("Merci de saisir votre prenom : ");
    string prenom = Console.ReadLine();
    string reponse = "HELLO " + nom + "\t" + prenom;
    Console.WriteLine(reponse);
}

static void M6()
```

```
static void TestIfv1()
{
    int a=15;
    int b = 10;
    if (a == b)
        Console.WriteLine("a et b egaux");
}
```

```
static void TestIfv2()
{
    int a = 10;
    int b = 10;
```



```

if (a == b)
    Console.WriteLine("a et b egaux");
else
    Console.WriteLine("a et b differents");
}

```

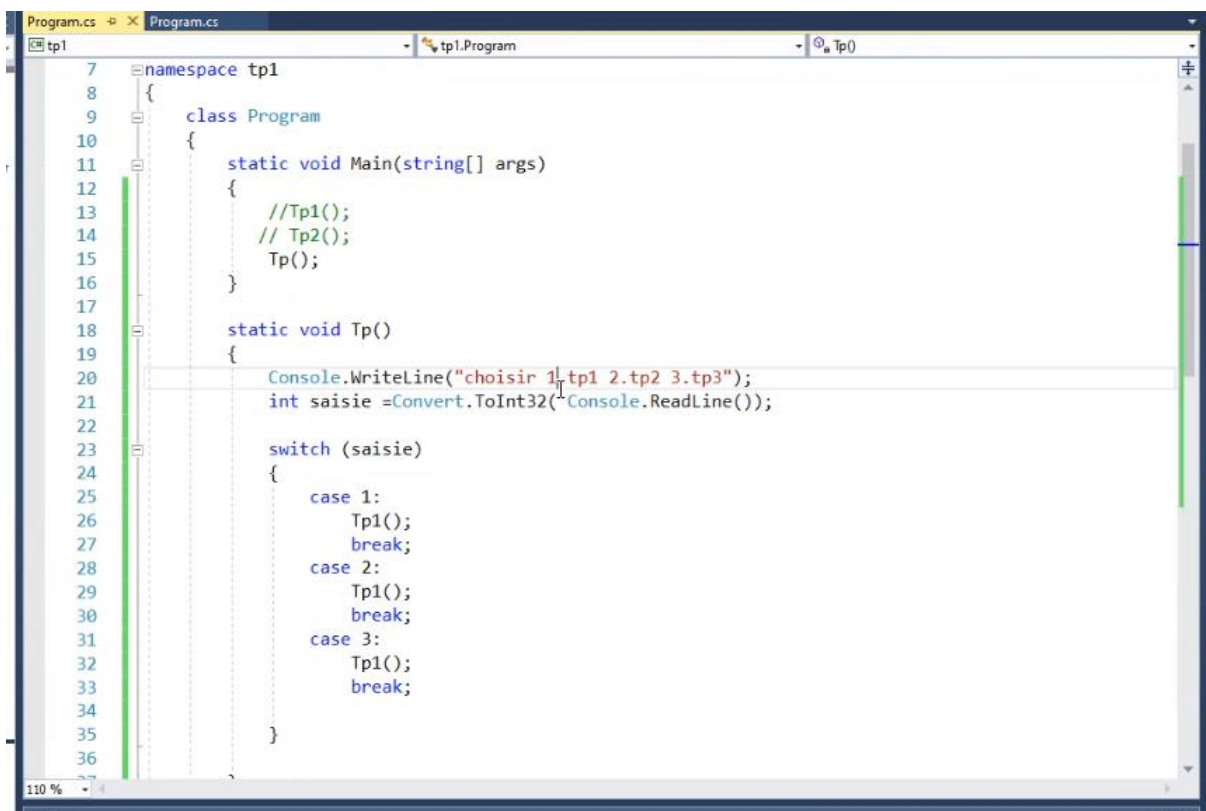
```

static void TestCase()
{
    string str1 = "toTO";
    string str2 = str1.ToLower();
    string str3 = str1.ToUpper();

    Console.WriteLine(str1);
    Console.WriteLine(str2);
    Console.WriteLine(str3);

    Console.WriteLine(str1.ToLower());
}

```



The image shows a Visual Studio IDE window with a C# program. The code is as follows:

```
11 static void Main(string[] args)
12 {
13     //Tp1();
14     // Tp2();
15     TpBoucleV1();
16 }
17
18
19 static void TpBoucleV1()
20 {
21     Console.WriteLine("Saisir la chaîne de caractère :");
22     string chaine = Console.ReadLine();
23
24     Console.WriteLine("Saisir la hauteur :");
25     int hauteur = Convert.ToInt32(Console.ReadLine());
26
27     Console.WriteLine("Saisir la largeur :");
28     int largeur = Convert.ToInt32(Console.ReadLine());
29
30     for (int i = 0; i < hauteur; i++)
31     {
32         for (int j = 0; j < largeur; j++)
33         {
34             Console.Write(chaine + "\t"); // Affiche la chaîne suivie d'une tabulation
35         }
36         Console.WriteLine(); // Saute une ligne à la fin de chaque ligne de largeur
37     }
38 }
39
40 }
```

The left sidebar shows the 'Exécuter...' menu and a list of tests. The top toolbar includes 'Debug', 'Any CPU', 'tp1', and 'Démarrer'.

```
1 using System;
2 using System.Text;
3
4 namespace ConsoleApp1
5 {
6     class Program
7     {
8
9         static void Main(string[] args)
10        {
11            TestPow();
12
13        }
14
15        static void TestStringBuilder()
16        {
17            StringBuilder sb = new StringBuilder();
18            sb.Append("bonjour");
19        }
20
21        static void TestPow()
22        {
23            double res = System.Math.Pow(2,3);
24            Console.WriteLine(res);
25        }
26
27        static void TestForv6()
28        {
29            for (int i = 0; i < 10; i++)
30            {
31                // ...
32            }
33        }
34    }
35 }
```

```
for (int i = 0; i < puissance; i++)
    reponse *= nb;
Console.WriteLine(reponse);
}

static void TpBoucleV3()
{
    Console.WriteLine("Veuillez entrer un caractère: ");
    string caract = Console.ReadLine();
    Console.WriteLine("Quelle hauteur ?");
    int x = Convert.ToInt32(Console.ReadLine());
    Console.WriteLine("Quelle largeur ?");
    int y = Convert.ToInt32(Console.ReadLine());
    StringBuilder sb = new StringBuilder();

    for (int i = 0; i < x; i++)
    {
        for (int j = 0; j < y; j++)
            sb.Append(caract + "\t");
        sb.Append("\n");
    }
    string resultat = sb.ToString();
    Console.WriteLine(resultat);
}

static void TpBoucleV2()
```

Build

la génération : Projet : Tp1, Configuration : Debug Any CPU -----

sktop\Cours\Cours_AJC\08_CSharp\CODE_SOURCE\SolutionTp\Tp1\Program.cs(218,20,218,21): warning CS0219: La variable

\greg\Desktop\Cours\Cours_AJC\08_CSharp\CODE_SOURCE\SolutionTp\Tp1\bin\Debug\Tp1.exe

on : 1 a réussi, 0 a échoué, 0 mis à jour, 0 a été ignoré

```
static void Addition(double a, double b)
{
    double c = a + b;
    Console.WriteLine(c);
}
```

```

main
    tpCalcul
    I

//user
tpCalcul()
saisir nb1,nb2,operateur

resultat = Calcul(nb1,nb2,operateur)
afficher resultat

//dev operation + - / *
// + add / div etc...

Calcul(nb1,nb2,operateur)
switch case
+ --> add retourne un resultat
/ --> div retourne un resultat
etc.....

add ....
div....
etc

```

```

{
    class Program
    {
        static void Main(string[] args)
        {
            TestPrint();
        }
        static void TestPrint()
        {
            Print();
            Print("toto");
        }
        static void Print()
        {
            Console.WriteLine("coucou");
        }
        //overloader surcharge de methodes
        static void Print(string nom)
        {
            Console.WriteLine("coucou "+nom);
        }
    }
}

```

```

//1
static void Print()
{
    Console.WriteLine("coucou");
}

//2.overloader surcharge de methodes
static void Print(string nom)
{

```

```

        Console.WriteLine("coucou NOM:"+nom);
    }

    //3
    static void Print(int a)
    {
        Console.WriteLine("coucou :" + a);
    }
    // 3 bis
    //static void Print(string x)
    //{
    //    Console.WriteLine("coucou NOM:" + x);
    //}

    //4
    static void Print( string a, int b)
    {
        Console.WriteLine("coucou " + a+" "+b);
    }
    //5
    static void Print(int a, string b)
    {
        Console.WriteLine("coucou " + a + " " + b);
    }
}
TpTableauv1
static void TestTableauIntv1()
{
    Console.WriteLine("Saisir la taille du tableau: ");
    int size = Convert.ToInt32(Console.ReadLine());

    string[] tab = new string[size];

    for(int i = 0; i < size; i++)
    {
        Console.WriteLine("Saisir l'element n°"+(i+1)+" du tableau:
(chaine de caracteres)");
        tab[i] = Console.ReadLine();
    }
    StringBuilder tableauAff = new StringBuilder();
    tableauAff.AppendLine("Elements du tableau: ");
    tableauAff.Append(tab[0]);
    for(int i = 1; i < size; i++)
    {
        tableauAff.Append("\t"+tab[i]);
    }
    tableauAff.Append("\n");
    Console.WriteLine(tableauAff);
}

```

Tpoccurrence

```
int[] tableau = { 10, 2, 30, 10, 1, 2, 5, 10 };  
Affiche(tableau);
```

```
Console.Write("Choisissez un entier : ");  
int choix = int.Parse(Console.ReadLine());
```

```
int compteur = Occurrence(tableau, choix);  
Console.WriteLine($"L'entier {choix} apparaît {compteur} fois dans le tableau.");
```

```
static void Affiche(int[] x)  
{  
    foreach (int e in x)  
        Console.Write(e + "\t");  
    Console.WriteLine();  
}
```

```
static int Occurrence(int[] tableau, int valeur)  
{  
    int compteur = 0;  
    foreach (int e in tableau)  
    {  
        if (e == valeur)  
            compteur++;  
    }  
    return compteur;  
}
```

tpTraitementText

```
static void tpTraitementTexte()  
    //{  
    //    Console.Write("Saisissez une phrase : ");  
    //    string phrase = Console.ReadLine();  
  
    //    string reponse = traitementTexte(phrase);  
  
    //    Console.WriteLine("Réponse : " + reponse);  
    //}  
  
    //static string traitementTexte(string str)  
    //{  
    //    string[] mots = str.Split(' ');  
    //    Array.Reverse(mots);  
    //    return String.Join(" ", mots).ToUpper();  
    //}
```