

Casse-têtes en C#

par Jon Skeet ([Page d'accueil](#)) ([Blog](#)) Jean-Michel Ormes (traduction) ([Page d'accueil](#)) ([Blog](#))

Date de publication : 11/04/2012

Dernière mise à jour :

Régulièrement, je tombe sur des situations intéressantes en C# qui amènent des résultats surprenants. Cette page contient une liste d'exemples. Pour ceux ne contenant qu'un bout de code, nous supposons que celui-ci est dans la méthode Main. Afin de ne pas tomber accidentellement sur les résultats avant que vous ne le souhaitiez, j'ai mis les **réponses sur une autre page**.

Commentez cet article :

Traduction.....	3
II - Surcharge.....	3
III - Ordre ! Ordre !.....	3
IV - Bête arithmétique.....	4
V - Print, print, print.....	4
VI - Il n'y a littéralement aucun problème avec le compilateur ici.....	4
VII - Inférence de type à gogo.....	5
VIII - Remerciements.....	5

Traduction

Ceci est la traduction la plus fidèle possible de l'article de Jon Skeet,  **C# Brainteasers**.

II - Surcharge

Qu'est-ce qui est affiché, et pourquoi ?

```
using System;

class Base
{
    public virtual void Foo(int x)
    {
        Console.WriteLine ("Base.Foo(int)");
    }
}

class Derived : Base
{
    public override void Foo(int x)
    {
        Console.WriteLine ("Derived.Foo(int)");
    }

    public void Foo(object o)
    {
        Console.WriteLine ("Derived.Foo(object)");
    }
}

class Test
{
    static void Main()
    {
        Derived d = new Derived();
        int i = 10;
        d.Foo(i);
    }
}
```

III - Ordre ! Ordre !

Qu'est-ce qui va s'afficher, pourquoi, et en êtes-vous sûr ?

```
using System;

class Foo
{
    static Foo()
    {
        Console.WriteLine ("Foo");
    }
}

class Bar
{
    static int i = Init();

    static int Init()
    {
        Console.WriteLine("Bar");
        return 0;
    }
}
```

```

    }
}

class Test
{
    static void Main()
    {
        Foo f = new Foo();
        Bar b = new Bar();
    }
}

```

IV - Bête arithmétique

Les ordinateurs sont censés être bons en calcul, n'est-ce pas ? Alors pourquoi la console renvoie-t-elle "False" ?

```

double d1 = 1.000001;
double d2 = 0.000001;
Console.WriteLine((d1-d2)==1.0);

```

V - Print, print, print...

Voici un code utilisant la fonctionnalité de méthode anonyme de C# 2. Que fait-il ?

```

using System;
using System.Collections.Generic;

class Test
{
    delegate void Printer();

    static void Main()
    {
        List<Printer> printers = new List<Printer>();
        for (int i=0; i < 10; i++)
        {
            printers.Add(delegate { Console.WriteLine(i); });
        }

        foreach (Printer printer in printers)
        {
            printer();
        }
    }
}

```

VI - Il n'y a littéralement aucun problème avec le compilateur ici...

Est-ce que ce code pourrait compiler ? Compile-t-il ? Qu'est-ce que cela signifie ?

```

using System;

class Test
{
    enum Foo { Bar, Baz };

    static void Main()
    {
        Foo f = 0.0;
        Console.WriteLine(f);
    }
}

```

```
}
```

En voici d'autres dans le même genre...

```
using System;

class Test
{
    enum Foo { Bar, Baz };

    const int One = 1;
    const int Une = 1;

    static void Main()
    {
        Foo f = One-Une;
        Console.WriteLine(f);
    }
}
```

VII - Inférence de type à gogo

J'ai d'abord vu ceci sur [le blog d'Ayende](#) (sous une forme un peu plus obscure, il est vrai). Encore une fois, réfléchissez sur ce que le code va afficher et pourquoi.

```
using System;

class Test
{
    static void Main()
    {
        Foo("Hello");
    }

    static void Foo(object x)
    {
        Console.WriteLine("object");
    }

    static void Foo<T>(params T[] x)
    {
        Console.WriteLine("params T[]");
    }
}
```

VIII - Remerciements

Je tiens à remercier Jon Skeet pour son aimable autorisation de traduire cet article, ainsi que [tomlev](#) et [ClaudeLELOUP](#) pour la relecture attentive et les corrections apportées.