

```
1  /*
2  Recursividade - função que chama a si mesma de forma direta (no mesmo bloco) ou
3  de forma indireta (no bloco seguinte). Pratica que deve ser evitado pois cria
4  muitos blocos de processamento na RAM, podendo causar overflow. Essa prática
5  é normalmente usada para lidar com estruturas de hierarquia, como arvores por exemplo.
6  */
7
8  class Program
9  {
10     static int fatorial(int n)
11     {
12         if (n == 0) return 1;
13         return (n * fatorial(n - 1));
14     }
15     static void Main(string[] args)
16     {
17         int num;
18         Console.Write("Digite um número: ");
19         num = int.Parse(Console.ReadLine());
20         Console.WriteLine("O fatorial de {0} é {1}", num, fatorial(num));
21         Console.ReadKey();
22     }
23 }
24
25 // Exemplo 2:
26
27 class Program
28 {
29     static void Exibe(int max, int[] v)
30     {
31         Console.WriteLine(v[max]);
32         if (max != v.Length - 1)
33         {
34             Exibe(max + 1, v);
35         }
36     }
37
38     static void Main(string[] args)
39     {
40         int[] vetor = new int[] { 10, 20, 30 };
41         Exibe(0, vetor);
42     }
43 }
```