Author

[dantx5](https://codefights.com/profile/dantx5)

https://codefights.com/img/coins_new.png2000

A *factorial number* is such a number that can be written as k! and is equal to the product of all integers 1through k. For example,

4! = 1 \* 2 \* 3 \* 4 = 24.

Any positive integer can be expressed as a sum of*factorial numbers*. Let the *factorial count* of a number nbe defined as the minimum number of *factorial numbers* required to result in a sum of n.

Given a positive integer n, return its *factorial count*.

**Example**

* For n = 8, the output should be  
  FactorialCount(n) = 2.

The *factorial count* of 8 is 2, because 8 = 3! + 2! = 6 + 2.

* For n = 145, the output should be  
  FactorialCount(n) = 3.

The *factorial count* of 145 is 3, because 145 = 5! + 4! + 1! = 120 + 24 + 1.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] integer n**

*Constraints:*  
1 ≤ n ≤ 4 · 106.

* **[output] integer**

The *factorial count* of n.

<https://codefights.com/challenge/2ZF7F2MPz4qTeyoBQ/main?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int factorial(int n)

{

int prod = 1;

for (int i = 2; i <= n; i++)

{

prod \*= i;

}

return prod;

}

static int FactorialCount(int n)

{

List<int> fact = new List<int>();

for (int i = 1; i <= 15; i++)

{

fact.Add(factorial(i));

}

foreach (int elem in fact)

{

Console.Write(elem + " ");

}

Console.WriteLine();

List<int> lista = new List<int>();

for (int i = fact.Count - 1; i >= 0; i--)

{

while (n >= fact[i])

{

lista.Add(fact[i]);

n -= fact[i];

}

}

foreach (int elem in lista)

{

Console.Write(elem + " ");

}

return lista.Count;

}

static void Main(string[] args)

{

Console.WriteLine("\nres: " + FactorialCount(723777));

Console.ReadLine();

}

}

}