Define the *alternating digit sum* of number "abc..xyz" as z - y + x - ... a, where a ... z are the digits the number consists of.

Given an integer n, your task is to find the *alternating digit sum* of n! and return it modulo 11.

**Example**

For n = 4, the output should be  
AlternatingDigitSum(n) = 2.

4! = 24, its *alternating digit* sum is equal to 2, and 2 % 11 = 2, which is the answer.

**Input/Output**

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

*Constraints*  
0 ≤ n ≤ 109.

* **[output] integer**

The *alternating digit sum* of n! modulo 11.

<https://codefights.com/challenge/uT2ywJR5QtJzJGqMb>

int factorial(int n)

{

int f = 1;

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

{

f \*= i;

}

return f;

}

int AlternatingDigitSum(int n)

{

if (n >= 11)

return 0;

string f = factorial(n).ToString();

int sum = 0, j = 0;

for (int i = f.Length - 1; i >= 0; i--)

{

if (j % 2 == 0)

{

sum += int.Parse(f[i].ToString());

}

else

{

sum -= int.Parse(f[i].ToString());

}

j++;

}

if (sum < 0)

return sum + 11;

return sum;

}