Author

[hưng\_n6](https://codefights.com/profile/h%C6%B0ng_n6)

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

You are given a number n. Count down from n to 1in accordance with the following condition:  
Each triplet but the first one should be reversed. If there're not enough numbers to make triplets, return"-1" instead.

**Examples:**

* HardNumber(9) = "987456123"
  + the first 3 numbers are 9, 8 and 7, so the countdown begins as "987";
  + the next 3 numbers are 6, 5, and 4. This tripled should be reversed, so the countdown continues to "987456";
  + the last triplet is 3, 2 and 1. This one should also be reversed, so the final answer is"987456123".
* HardNumber(10) = -1
  + the first 3 numbers are 10, 9 and 8;
  + the next triplet is 7, 6, 5;
  + the next one is 4, 3, 2;
  + finally, there's only 1 left, which doesn't belong to any triplet.
* **[input] integer n**
  + The number to start the countdown from, 1 ≤ n ≤ 1500.
* **[output] string**
  + The countdown.

<https://codefights.com/challenge/KBXfWrndEuZ6fY2Wf>

static string countDown(int n)

{

bool es\_primero = true;

string answer = "";

int i = n;

while (i >= 1)

{

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

for (int cont = 0; cont < 3; cont++)

{

if (i >= 1)

{

//Console.Write(i);

triplet.Add(i);

}

else if (i <= 0)

{

return "-1";

}

i--;

}

string rev = "";

for (int j = triplet.Count - 1; j >= 0; j--)

{

rev += triplet[j].ToString();

}

string trip = "";

for (int j = 0; j < triplet.Count; j++)

{

trip += triplet[j].ToString() ;

}

if (es\_primero)

{

answer += trip;

es\_primero = false;

}

else

{

answer += rev;

}

// Console.WriteLine(rev);

// Console.WriteLine(trip);

// Console.WriteLine();

}

return answer;

}