You and your friends love to play bowling at the end of a long week. You play by your very own rules. Each team has an even number of *rolls* to make, two *rolls* in a row. A pair of consecutive rolls is called a *frame*.

Each frame 10 pins are placed on a lane. If a player manages to knock down all 10 pins, he gets 30 points, and can try to knock down another 10 pins in the next *roll*. If he knocks them down again, he gets another 30 points. Otherwise the number of points he gains is equal to the number of pins he knocked down. If a player doesn't knock down all the pins in the first*roll* of a *frame*, by the end of the *frame* he gets the number of points equal to the number of pins knocked down in that *frame*.

You are given the results of the rolls your team played. Return the total number of points you received this game.

**Example**

For rolls = [10, 10, 10, 4, 9, 1, 0, 10, 3, 3, the output should be  
bowlingScore(rolls) = 120.

Here's the number of points you got in each*frame*:

1. 30 + 30 = 60
2. 30 + 4 = 34
3. 9 + 1 = 10
4. 0 + 10 = 10
5. 3 + 3 = 6.

Thus, the answer is 60 + 34 + 10 + 10 + 6 = 120.

**Input/Output**

* **[time limit] 3000ms (cs)**
* **[input] array.integer rolls**

An array of integers. The length of the array is guaranteed to be even.

*Constraints:*  
2 ≤ rolls.length ≤ 20,  
0 ≤ rolls[i] ≤ 10.

* **[output] integer**

The total score by the end of the game.

<https://codefights.com/challenge/5vqcGLGeHmaC5mxCG?utm_source=featuredChallenge&utm_medium=email&utm_campaign=email_notification>

static int bowlingScore(int[] rolls)

{

int puntos = 0;

int i;

for (i = 0; i < rolls.Length; i += 2)

{

if (rolls[i] == 10)

{

puntos += 30;

if (rolls[i + 1] == 10)

{

puntos += 30;

}

else

{

puntos += rolls[i + 1];

}

}

else

{

puntos += rolls[i];

if(i+1 <rolls.Length) {

puntos += rolls[i + 1];

}

}

// puntos += rolls[i + 1];

}

return puntos;

}