Tom goes to a train station every afternoon to watch the 2pm train go. He has noted that the number of wagons the train has is always an odd number, and that every day the number of wagons the train has increases by 2, starting with 1 wagon on the first day of each month.

Thus, on the 1st day of each month the train goes with a single wagon, on the 2nd day it has 3 wagons, the next day it has 5 wagons, and so on.

The routine of counting the wagons comforts Tom, and he never misses a day. But on the dayth day of a certain month he will have to go to a camp that will last for n days. During these days, Tom will not be able to come and count the wagons.

Given the day, the month and the number n, your task is to calculate the number of wagons Tom will miss. Assume that this and the following years are both non-leap.

**Example**

* For month = 1, day = 1 and n = 1, the output should be  
  toms\_wagons(month, day, n) = 1.

Tom will miss only the train that leaves on the first of January. This train will have only one wagon, so the answer is 1.

* For month = 3, day = 2 and n = 4, the output should be  
  toms\_wagons(month, day, n) = 24.

Tom will miss 4 March days, from the 2nd to the 5th. Thus, he will miss 3 + 5 + 7 + 9 = 24wagons.

**Input/Output**

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

The 1-based month of the year.

*Constraints:*  
1 ≤ month ≤ 12.

* **[input] integer day**

The day of the month when the camp starts.

*Constraints:*  
1 ≤ day ≤ *the\_number\_of\_days\_in\_the\_ith\_ month*.

* **[input] integer n**

The number of days Tom will spend in the camp (equal to the number of trains he will miss).

*Constraints:*  
0 ≤ n ≤ 365.

* **[output] integer**

The number of wagons Tom will miss.

<https://codefights.com/challenge/X8dWNNGxbAAnMkBaB>

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

namespace ConsoleApplication1

{

class Program

{

static int toms\_wagons(int month, int day, int n)

{

int[] meses = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };

int days = 0;

int dia = day;

int mes = month-1;

int vagon = 0;

int impar =1;

for (int i = 0; i < dia; i++)

{

vagon = impar;

impar += 2;

}

int ans = 0;

for (int i = 0; i < n; i++)

{

//Console.Write(dia + " ");

Console.Write(vagon + " ");

dia++;

ans += vagon;

vagon += 2;

if (dia > meses[mes])

{

dia = 1;

vagon = 1;

mes++;

}

if (mes > 11)

{

mes = 0;

}

}

return ans;

}

static void Main(string[] args)

{

int month = 3, day = 2, n = 4;

// int month = 3, day = 2, n = 35;

Console.WriteLine( toms\_wagons(month, day, n));

Console.ReadLine();

}

}

}