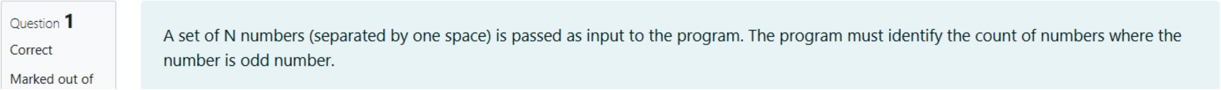
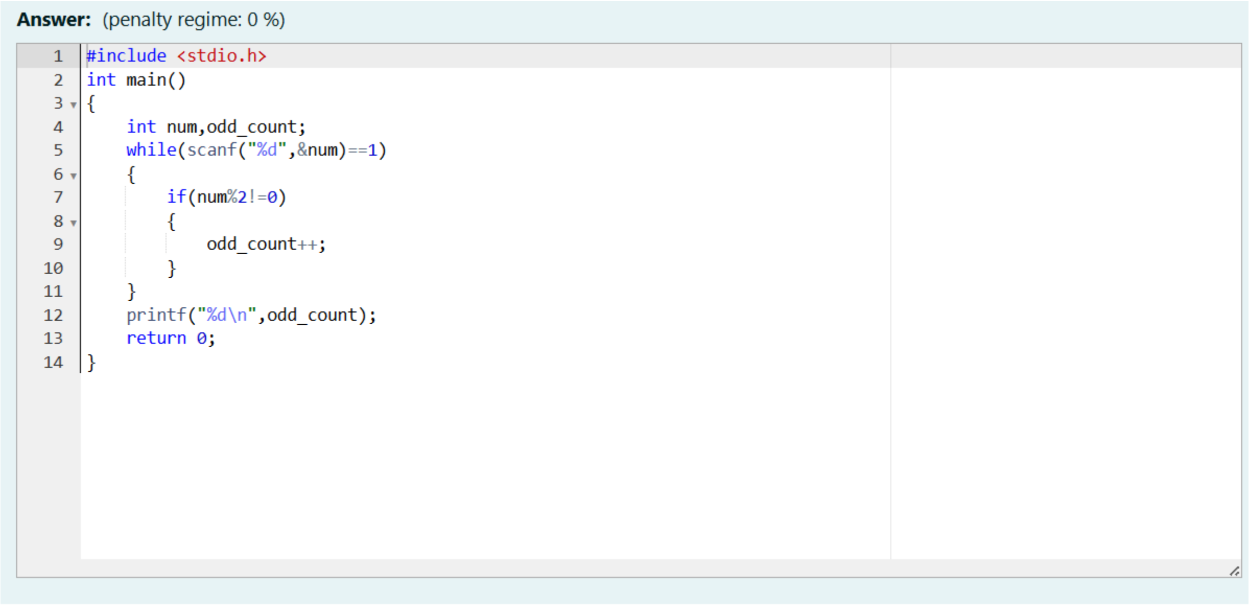


Question **1**

Correct

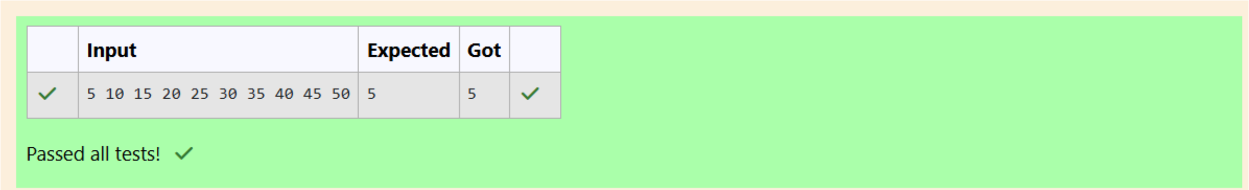
Marked out of

A set of N numbers (separated by one space) is passed as input to the program. The program must identify the count of numbers where the number is odd number.



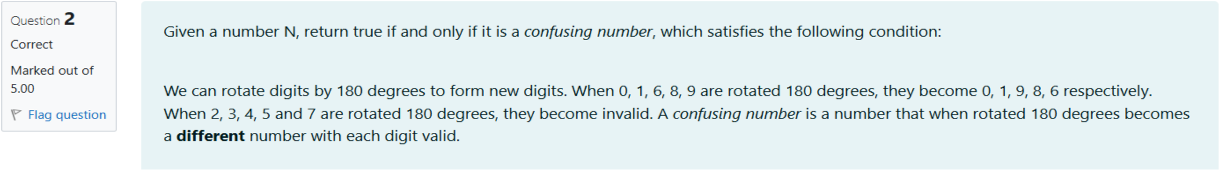
Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int num,odd_count;
    while(scanf("%d",&num)==1)
    {
        if(num%2!=0)
        {
            odd_count++;
        }
    }
    printf("%d\n",odd_count);
    return 0;
}
```



	Input	Expected	Got	
✓	5 10 15 20 25 30 35 40 45 50	5	5	✓

Passed all tests! ✓



Question **2**

Correct

Marked out of 5.00

Flag question

Given a number N, return true if and only if it is a *confusing number*, which satisfies the following condition:

We can rotate digits by 180 degrees to form new digits. When 0, 1, 6, 8, 9 are rotated 180 degrees, they become 0, 1, 9, 8, 6 respectively. When 2, 3, 4, 5 and 7 are rotated 180 degrees, they become invalid. A *confusing number* is a number that when rotated 180 degrees becomes a **different** number with each digit valid.

Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int num,d;
    scanf("%d",&d);
    num = d%10;
    if(d==11)
    {
        printf("false");
    }
    else if(num==0 || num==1 ||  num==6 || num==8 || num==9)
    {
        printf("true");
    }
    else
    {
        printf("false");

    }
    return 0;
}
```

	Input	Expected	Got	
✓	6	true	true	✓
✓	89	true	true	✓
✓	25	false	false	✓

Passed all tests! ✓

Question **3**

Correct

Marked out of 7.00

Flag question

A nutritionist is labeling all the best power foods in the market. Every food item arranged in a single line, will have a value beginning from 1 and increasing by 1 for each, until all items have a value associated with them. An item's value is the same as the number of macronutrients it has. For example, food item with value 1 has 1 macronutrient, food item with value 2 has 2 macronutrients, and incrementing in this fashion.

The nutritionist has to recommend the best combination to patients, i.e. maximum total of macronutrients. However, the nutritionist must avoid prescribing a particular sum of macronutrients (an 'unhealthy' number), and this sum is known. The nutritionist chooses food items in the increasing order of their value. Compute the highest total of macronutrients that can be prescribed to a patient, without the sum matching the given 'unhealthy' number.

Here's an illustration:

Given *4* food items (hence value: *1,2,3* and *4*), and the unhealthy sum being *6* macronutrients, on choosing items *1, 2, 3* -> the sum is *6*, which matches the 'unhealthy' sum. Hence, one of the three needs to be skipped. Thus, the best combination is from among:

- *2 + 3 + 4 = 9*
- *1 + 3 + 4 = 8*
- *1 + 2 + 4 = 7*

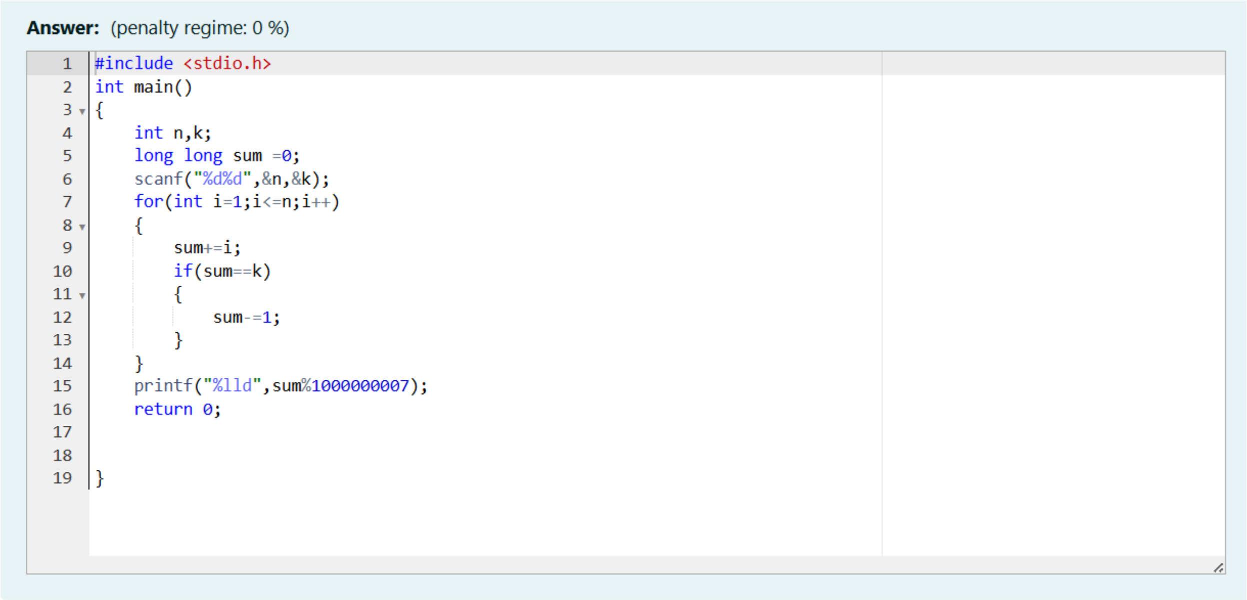
Since *2 + 3 + 4 = 9*, allows for maximum number of macronutrients, 9 is the right answer.

Complete the code in the editor below. It must return an integer that represents the maximum total of macronutrients, modulo *1000000007* *(10^9 + 7)*.

It has the following:

n: an integer that denotes the number of food items

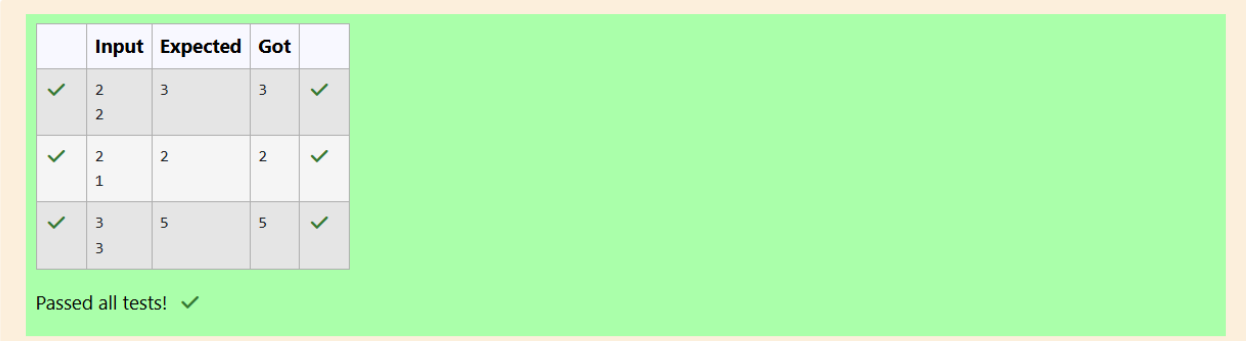
k: an integer that denotes the unhealthy number



Answer: (penalty regime: 0 %)

```
#include <stdio.h>
int main()
{
    int n,k;
    long long sum =0;
    scanf("%d%d",&n,&k);
    for(int i=1;i<=n;i++)
    {
        sum+=i;
        if(sum==k)
        {
            sum-=1;
        }
    }
    printf("%lld",sum%1000000007);
    return 0;


}
```



	Input	Expected	Got	
✓	2 2	3	3	✓
✓	2 1	2	2	✓
✓	3 3	5	5	✓

Passed all tests! ✓