Status	Finished
Started	Saturday, 1 November 2025, 8:47 PM
Completed	Saturday, 1 November 2025, 9:39 PM
Duration	51 mins 56 secs

Question **1**

Correct

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.

Input Format:

The first line will contain the N numbers separated by one space.

Boundary Conditions:

3 <= N <= 50

The value of the numbers can be from -99999999 to 99999999

Output Format:

The count of numbers where the numbers are odd numbers.

Example Input / Output 1:

Input:

5 10 15 20 25 30 35 40 45 50

Output:

5

Explanation:

The numbers meeting the criteria are 5, 15, 25, 35, 45.

Answer: (penalty regime: 0 %)

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

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

Passed all tests! ✓

Question **2**

Correct

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.

Example 1:

6 -> 9

Input:

6

Output:

true

Explanation:

We get 9 after rotating 6, 9 is a valid number and 9!=6.

Example 2:

89 -> 68

Input:

89

Output:

true

Explanation:

We get 68 after rotating 89, 86 is a valid number and 86!=89.

Example 3:

11 -> 11

Input:

11

Output:

false

Explanation:

We get 11 after rotating 11, 11 is a valid number but the value remains the same, thus 11 is not a confusing number.

Note:

1. 0 <= N <= 10^9
2. After the rotation we can ignore leading zeros, for example if after rotation we have 0008 then this number is considered as just 8.

Answer: (penalty regime: 0 %)

```c
#include <stdio.h>
int main(){
    long long n, temp, rotated = 0;
    int digit, valid = 1;
    scanf("%lld", &n);
    temp = n;
    while(temp > 0){
        digit = temp % 10;
        if(digit == 2 || digit == 3 || digit == 4 || digit == 5 || digit
            valid = 0;
            break;
        }
        switch (digit){
            case 0: rotated = rotated * 10 + 0; break;
            case 1: rotated = rotated * 10 + 1; break;
            case 6: rotated = rotated * 10 + 9; break;
            case 8: rotated = rotated * 10 + 8; break;
            case 9: rotated = rotated * 10 + 6; break;
        }
        temp /=10;
    }
    if (!valid){
        printf("false");
        return 0;
    }
    if (rotated != n)
        printf("true");
    else
        printf("false");
    return 0;
}
```

```
31 }
```

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

Passed all tests! ✓

Question **3**

Correct

Problem Statement:
In a small data-entry office, the operator records daily temperature changes in a city.
Positive numbers indicate temperature rises, and negative numbers indicate temperature drops.
The operator enters all temperature changes for a day on a single line, separated by spaces.
Once the operator presses Enter, the program must calculate and display:
The total count of positive temperature changes.
The total count of negative temperature changes.
This will help the office track how many times the temperature rose or fell during the day.

Boundary Conditions:
The number of integers entered can vary.
Each integer can range from -99999999 to 99999999.

Sample Input:
5 -2 10 0 -3 7
Sample Output:
Positive numbers count: 3
Negative numbers count: 2

Answer: (penalty regime: 0 %)

```c
#include<stdio.h>
int main()
{
    int num;
    int positivecount = 0, negativecount = 0;
    while(scanf("%d", &num) == 1)
    {
        if (num >= 0)
            positivecount++;
        else if (num < 0)
            negativecount++;
    }
    printf("Positive numbers count: %d\n",positivecount);
    printf("Negative numbers count: %d\n",negativecount);
    return 0;
}
```

	Input	Expected	Got	
✓	-10 5 6 -6	Positive numbers count: 2 Negative numbers count: 2	Positive numbers count: 2 Negative numbers count: 2	✓
✓	10 -6 6 7 8 9 4 0	Positive numbers count: 7 Negative numbers count: 1	Positive numbers count: 7 Negative numbers count: 1	✓

Passed all tests! ✓