

Imigongo Artworks

Imigongo is a traditional Rwandan art form known for its beautiful geometric patterns. An Imigongo workshop is preparing several artworks for an exhibition in Kigali. The artworks have been placed inside a storage chest to protect them during transportation. However, when it is time to leave, the artisans discover that nobody remembers the code needed to open the chest.

The chest has a lock displaying N decimal digits. The artisans cannot change any of the lock's digits. They can only press a button that moves the rightmost digit to the leftmost position.

For example, pressing the button once changes 12345 to 51234 and pressing it twice changes it to 45123.

After pressing the button, the leftmost digit may be 0. Such zeroes remain on the display and are included when determining the positions of the digits. However, they do not change the value of the represented integer. For example, pressing the button once changes 120 into 012 which represents the integer 12 and pressing the button twice changes 120 to 201.

The artisans remember that the chest opens whenever the displayed number on the lock is divisible by 11.

An integer is divisible by 11 if and only if the alternating sum of its digits is divisible by 11.

Suppose that the displayed digits from left to right are $D[0], D[1], \dots, D[N-1]$. Their alternating sum is obtained by adding the digits in positions 0, 2, 4, ... and subtracting the digits in positions 1, 3, 5, ... In other words, it is $D[0] - D[1] + D[2] - D[3] + \dots$. The represented integer is divisible by 11 when the alternating sum is divisible by 11.

For example, 86834 is divisible by 11 because $8 - 6 + 8 - 3 + 4 = 11$ which is divisible by 11. Meanwhile 11111 is not divisible by 11, because $1 - 1 + 1 - 1 + 1 = 1$, which is not divisible by 11.

You need to help the artisans by determining the minimum number of button presses (possibly zero) needed to obtain an integer divisible by 11 or tell them that it is impossible to achieve an integer divisible by 11 using any number of button presses.

Implementation Details

You should implement the following procedure:

```
int32 count_presses(string D)
```

Let N be the length of D .

- D : a string containing the digits of the displayed integer with $D[0]$ being the leftmost digit and $D[N - 1]$ being the rightmost digit.
- This procedure should return the minimum number of button presses to make the displayed integer divisible by 11 or -1 if it's impossible to obtain an integer divisible by 11 using any number of button presses.
- This procedure may be called up to 100 times.

C interface

For C submissions, include the provided header `imigongo_c.h` and implement:

```
int count_presses(char *D);
```

In the C interface, D is a null-terminated character array owned by the grader. Your procedure must not free it.

Constraints

- $1 \leq N \leq 10^5$
- Each character of D is a digit between "0" and "9".
- The sum of N over all calls to `count_presses` does not exceed 10^5 .

Subtasks

Subtask	Score	Additional Constraints
1	5	$N = 1$
2	5	$N = 2$
3	8	$N \leq 8$ and N is even.
4	14	$N \leq 17$ and N is even.
5	23	N is even.
6	26	$N \leq 1000$
7	19	No additional constraints.

Examples

Example 1

Consider the following call.

```
count_presses("82274")
```

In this example, $D = "82274"$.

After one button press, the display becomes "48227", which represents 48227 and is not divisible by 11. After two button presses, the display becomes "74822". Its alternating sum is $7 - 4 + 8 - 2 + 2 = 11$, so it is divisible by 11. Therefore, the procedure should return 2.

Example 2

Consider the following call.

```
count_presses("1234")
```

In this example, $D = "1234"$.

In this example, no number obtainable by pressing the button is divisible by 11. Therefore, the procedure should return -1 .

Example 3

Consider the following call.

```
count_presses("11")
```

In this example, $D = 11$.

Since the displayed integer is already divisible by 11, we do not need to press the button. Therefore, the procedure should return 0.

Sample Grader

Input format:

The first line of the input contains an integer T , specifying the number of scenarios. T scenarios follow. Each of them is provided in the following format:

D

Output format:

The sample grader writes the result of each of the T scenarios in the same order as they are provided in the input in the following format:

C

Here, C is the return value of `count_presses`.