Codeforces Round 904 (Div. 2)

B. Haunted House

1 second, 256 megabytes

You are given a number in binary representation consisting of exactly n bits, possibly, with leading zeroes. For example, for $n = 5$ the number 6 will be given as 00110, and for $n = 4$ the number 9 will be given as 1001.

Let's fix some integer i such that $1 \le i \le n$. In one operation you can swap any two adjacent bits in the binary representation. Your goal is to find the smallest number of operations you are required to perform to make the number divisible by 2^i, or say that it is impossible.

Please note that for each $1 \le i \le n$ you are solving the problem independently.

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^4)$. The description of the test cases follows.

The first line of each test case contains one integer n $(1 \le n \le 10^5)$ — the length of the binary representation of the number.

The second line of each test case contains a string of length n, consisting of 0 and 1, — the binary representation of the number.

It is guaranteed that the sum of n over all test cases does not exceed $2 \cdot 10^5$.

Output

For each test case, for each $1 \le i \le n$ output the smallest number of operations required to make the number divisible by 2^i, or output -1 if it is impossible.

input

```
6
1
1
2
01
3
010
5
10101
7
0000111
12
001011000110
```

output

```
-1
1 -1
0 1 -1
1 3 -1 -1 -1
3 6 9 12 -1 -1 -1
0 2 4 6 10 15 20 -1 -1 -1 -1 -1
```

In the first test case, we cannot swap any elements, and the number 1 is not divisible by 2.

In the second test case, the initial number is 1. It is not divisible by 2, but if we perform the operation, then we obtain the number with binary representation 10, which is equal to 2 in decimal representation, thus, it is divisible by 2. But this number is not divisible by 4 and we cannot obtain any other number using the operations.

In the third test case, the initial number is 2. For $i = 1$ we do not have to perform any operations since the initial number is divisible by 2. For $i = 2$ we can perform one operation swapping the first two bits (we would obtain 100 in binary representation, which corresponds to number 4). There is no answer for $i = 3$.