Flag Game

Time Limit: 2 sec / Memory Limit: 1024 MB
Score : 200 points

PROBLEM STATEMENT

We have **L** exun flags, **M** cw flags, and **N** ts flags. You can do the following operation any number of times:

- choose two different flags and turn them into the remaining flag.

For example, you can choose a cw flag and a ts flag and turn them into two exun flags.

Your objective is to convert all the flags into the flag of a particular club. Determine whether this objective is achievable. If it is, find the minimum number of operations required to achieve it.

For each input file, solve ***T*** test cases.

CONSTRAINTS

- $1 \le T \le 100$
- $1 \le L,M,N \le 10^8$
- All values in input are integers.

INPUT FORMAT

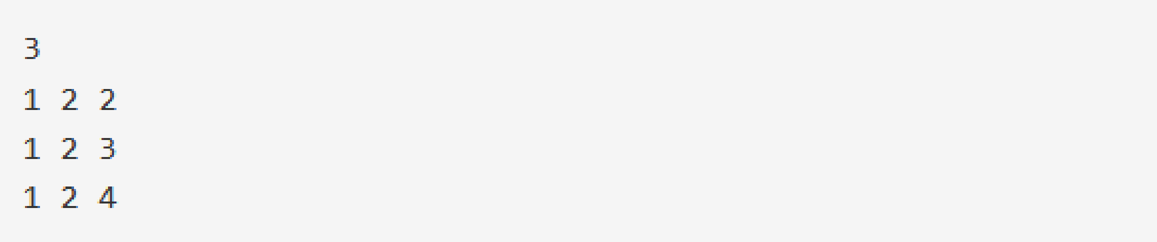
The first line contains an integer- T (number of test cases)

The next T lines contain 3 space separated integers- L, M, N

OUTPUT FORMAT

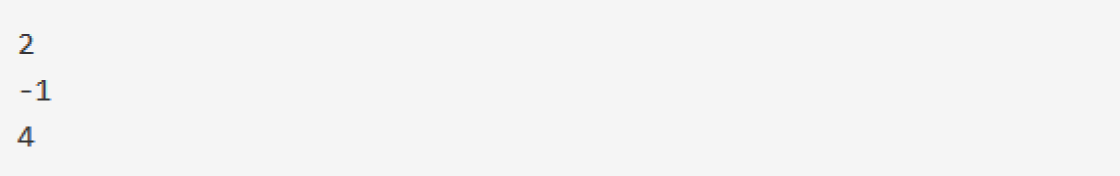
For each case, print **-1** if the objective is unachievable; otherwise, print the **minimum** number of operations to achieve it.

SAMPLE INPUT 1



```
3
1 2 2
1 2 3
1 2 4
```

SAMPLE OUTPUT 1



```
2
-1
4
```

For example, in case 3, one optimal sequence of operations is:

- choose a cw flag and ts flag, turning them into two exun flags;
- choose a exun flag and ts flag, turning them into two cw flags;
- choose a exun flag and ts flag, turning them into two cw flags;
- choose a exun flag and ts flag, turning them into two cw flags;

Color Grid

Time Limit: 2 sec / Memory Limit: 1024 MB
Score : 400 points

PROBLEM STATEMENT

We have a H×W grid. Initially, the squares are unpainted.

You will paint these squares. There are C colors available, numbered 1,2,...,C.

The painting process will be given as Q queries. The i-th query contains integers t_i , n_i , c_i , which represents the following action.

- If t_i=1: paint all squares in the n_i-th **row** with Color c_i .
- If t_i=2: paint all squares in the n_i-th **column** with Color c_i .

Painting a square with Color c makes the color of that square Color c, regardless of its previous state.

Find the number of squares painted in each of Color 1,2,...,C after the whole process.

CONSTRAINTS

- $2 \le H \le 109$
- $2 \le W \le 109$
- $1 \le C \le 3\times105$
- $1 \le Q \le 3\times105$
- $t_i \in \{1,2\}$
- $1 \le n_i \le H$ if $t_i = 1$
- $1 \le n_i \le W$ if $t_i = 2$
- $1 \le c_i \le C$

INPUT FORMAT

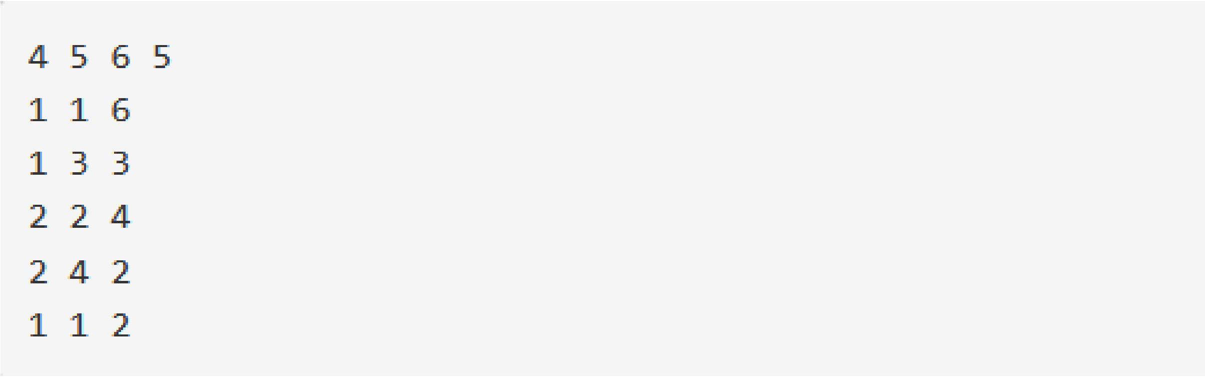
The first line contains 4 space separated integers- H, W, C, Q

The next Q lines contain 3 space separated integers- $t_{i,}$ $n_{i,}$ c_i

OUTPUT FORMAT

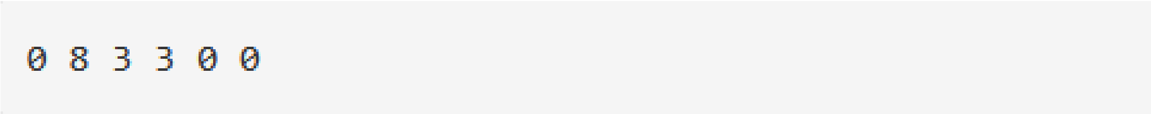
Print a line containing the numbers of squares painted in Color 1 , 2 , ... , C with spaces in between.

SAMPLE INPUT 1



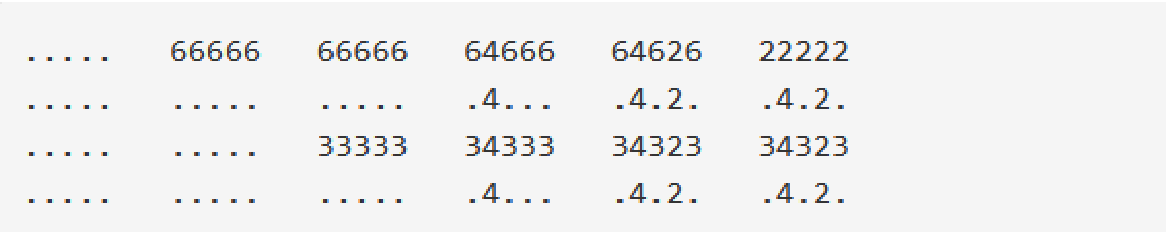
```
4 5 6 5
1 1 6
1 3 3
2 2 4
2 4 2
1 1 2
```

SAMPLE OUTPUT 1



```
0 8 3 3 0 0
```

The process changes the colors of the squares as follows. Here, '.' denotes an unpainted square.



```
.....   66666   66666   64666   64626   22222
.....   .....   .....   .4...   .4.2.   .4.2.
.....   .....   33333   34333   34323   34323
.....   .....   .....   .4...   .4.2.   .4.2.
```

C.O.R.E. Challenges Prez

Time Limit: 2 sec / Memory Limit: 1024 MB
Score : 600 points

PROBLEM STATEMENT

C.O.R.E. members challenged their president to solve a math riddle. Manan, wanting to prove himself, accepted the challenge. They gave Manan integers a and b. Initially, a = 0, b = 0.

Manan can do the following four operations any number of times in any order:

- Operation 1: Replace the value of a with a+1.
- Operation 2: Replace the value of b with b+1.
- Operation 3: Replace the value of a with a+b.
- Operation 4: Replace the value of b with a+b.

SInce Manan is very weak in math, he called you to help him win the challenge. You are given a positive integer ***N***. Do at most ***130*** operations so that 'a' will have the value ***N***. Here, b can have any value. C.O.R.E. members can prove that such a sequence of operations exists under the constraints of this question.

CONSTRAINTS

- $1 \leq N \leq 10^{18}$
- All values in input are integers.

INPUT FORMAT

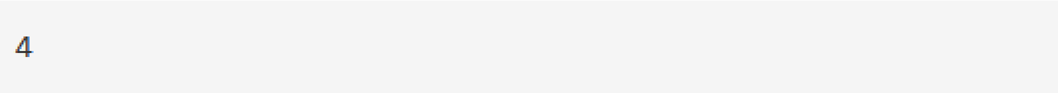
The first and only line contains an integer: ***N***

OUTPUT FORMAT

The first line outputs the number of operations performed: ***K***

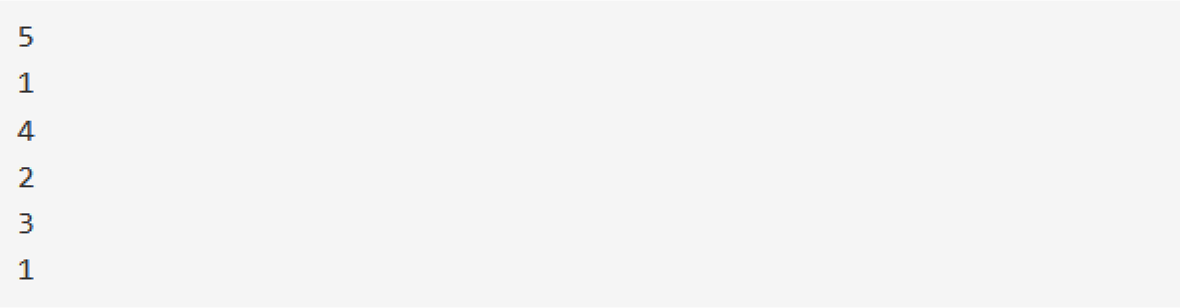
The next ***K*** lines output the operation number i.e. 1, 2, 3 or 4.

SAMPLE INPUT 1



```
4
```

SAMPLE OUTPUT 1



```
5
1
4
2
3
1
```

Here, the values of a and b change as follows:

(0,0)→ (Operation 1) → (1,0) → (Operation 4) → (1,1) → (Operation 2) → (1,2) → (Operation 3) → (3,2) → (Operation 1) → (4,2),

and the final value of a matches N.

C.O.R.E. Dance

Time Limit: 2 sec / Memory Limit: 1024 MB
Score : 1000 points

PROBLEM STATEMENT

There are ***3^N*** C.O.R.E. members dancing in a circle. We denote with 0 , 1 , ... , 3^N–1 the positions in the circle, starting from an arbitrary position and going around clockwise. Initially each position in the circle is occupied by one member.

The members are going to dance on two kinds of songs: Exun anthem and CW anthem.

- When the Exun anthem is played, the member in position i goes to position *j*, where *j* is the number obtained replacing all digits 1 with 2 and all digits 2 with 1 when reading *i* in base 3 (e.g., the person in position 46 goes to position 65).
- When the CW anthem is played, the member in position *i* moves to position *i+1* (with the identification $3^N = 0$).

Manan is given a string $T=T_1T_2 \cdots T_{|T|}$ such that T_i = **E** if the *i*-th song is Exun anthem and T_i = **C** if it is CW anthem. After all anthems have been played, the member that initially was in position i is in position P_i . Compute the array P_1 , P_2 ..., P_{3^N-1}.

CONSTRAINTS

- 1 ≤ N ≤ 12
- 1 ≤ ∣T∣ ≤ 200,000
- T contains only the characters E and C.

INPUT FORMAT

The first line contains an integer- N

The next line contains the string- T

OUTPUT FORMAT

$$P_0 \quad P_1 \quad \cdots \quad P_{3^N-1}$$

SAMPLE INPUT 1

```
1
ECE
```

SAMPLE OUTPUT 1

```
2 0 1
```

Before any anthem is played, the positions are: 0, 1, 2.

When we say "member ***i***", we mean "the member that was initially in position ***i***".

1. After the first exun anthem, the positions are: 0, 2, 1.
2. After the cw anthem, the positions are: 1, 0, 2 (so, member 0 is in position 1, member 1 is in position 0 and member 2 is in position 2).
3. After the second exun anthem, the positions are: 2, 0, 1 (so, member 0 is in position 2, member 1 is in position 0 and member 2 is in position 1).

C.O.R.E. is Hungry

Time Limit: 2 sec / Memory Limit: 1024 MB
Score : 1000 points

PROBLEM STATEMENT

There are ***N*** cupcakes kept in a row in Prakash Bakery, each of which has its price written on it. The price written on the ***i-th*** cupcake from the left is $\boldsymbol{A_i}$.

C.O.R.E. will repeat the following operation until two cupcakes remain:

- Choose three consecutive cupcakes from the row.
- Eat the middle cupcake of the three.
- For each of the other two cupcakes, replace the price written on it by the sum of the price of that cupcake and the price written on the cupcake eaten.
- Return the two cupcakes to the original position in the row, without swapping them.

Find the minimum possible sum of prices written on the last two cupcakes remaining.

CONSTRAINTS

- $2 \leq N \leq 18$
- $0 \leq A_i \leq 10^9$
- All values in input are integers.

INPUT FORMAT

The first line contains an integer- N

The next line contains N space separated integers - A_i

OUTPUT FORMAT

Print the minimum possible sum of prices written on the last two cupcakes remaining.

SAMPLE INPUT 1

```
4
3 1 4 2
```

SAMPLE OUTPUT 1

```
16
```

We can minimize the sum of prices written on the last two cupcakes remaining by doing as follows:

- Initially, the price written on the cupcakes are 3, 1, 4, and 2 from left to right.

- Choose the first, second, and third cupcake from the left. Eat the second cupcake whose price is 1, add 1 to the prices of each of the other two cupcakes, and return them to the original position in the row. The prices written on the cupcakes are now 4, 5, and 2 from left to right.

- Choose the first, second, and third cupcake from the left. Eat the second cupcake whose price is 5, add 5 to the price of each of the other two cupcakes, and return them to the original position in the row. The integers written on the cupcakes are now 9 and 7 from left to right.

- The sum of the prices written on the last two cupcakes remaining is 16.