Sponsored by Telegram

Codeforces Round #676 (Div. 2)

A. XORwice

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

In order to celebrate Twice's 5th anniversary, Tzuyu and Sana decided to play a game.

Tzuyu gave Sana two integers a and b and a really important quest.

In order to complete the quest, Sana has to output the smallest possible value of ($a \oplus x$) + ($b \oplus x$) for any given x, where $\oplus$ denotes the bitwise XOR operation.

Input

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

The only line of each test case contains two integers a and b ($1 \le a, b \le 10^9$).

Output

For each testcase, output the smallest possible value of the given expression.

Example

input

```
6
6 12
4 9
59 832
28 14
4925 2912
1 1
```

output

```
10
13
891
18
6237
0
```

Note

For the first test case Sana can choose $x = 4$ and the value will be ($6 \oplus 4$) + ($12 \oplus 4$) = $2 + 8$ = 10. It can be shown that this is the smallest possible value.

B. Putting Bricks in the Wall

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Pink Floyd are pulling a prank on Roger Waters. They know he doesn't like walls, he wants to be able to walk freely, so they are blocking him from exiting his room which can be seen as a grid.

Roger Waters has a square grid of size $n \times n$ and he wants to traverse his grid from the upper left $(1, 1)$ corner to the lower right corner (n, n). Waters can move from a square to any other square adjacent by a side, as long as he is still in the grid. Also except for the cells $(1, 1)$ and (n, n) every cell has a value 0 or 1 in it.

Before starting his traversal he will pick either a 0 or a 1 and will be able to only go to cells values in which are equal to the digit he chose. The starting and finishing cells $(1, 1)$ and (n, n) are exempt from this rule, he may go through them regardless of picked digit. Because of this the cell $(1, 1)$ takes value the letter 'S' and the cell (n, n) takes value the letter 'F'.

For example, in the first example test case, he can go from $(1, 1)$ to (n, n) by using the zeroes on this path: $(1, 1)$, $(2, 1)$, $(2, 2)$, $(2, 3)$, $(3, 3)$, $(3, 4)$, $(4, 4)$

The rest of the band (Pink Floyd) wants Waters to not be able to do his traversal, so while he is not looking they will **invert at most two cells** in the grid (from 0 to 1 or vice versa). They are afraid they will not be quick enough and asked for your help in choosing the cells. **Note that you cannot invert cells $(1, 1)$ and (n, n).**

We can show that there always exists a solution for the given constraints.

Also note that Waters will pick his digit of the traversal after the band has changed his grid, so he must not be able to reach (n, n) no matter what digit he picks.

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 50$). Description of the test cases follows.

The first line of each test case contains one integers n ($3 \le n \le 200$).

The following n lines of each test case contain the binary grid, square $(1, 1)$ being colored in 'S' and square (n, n) being colored in 'F'.

The sum of values of n doesn't exceed 200.

Output

For each test case output on the first line an integer c ($0 \le c \le 2$) — the number of inverted cells.

In i-th of the following c lines, print the coordinates of the i-th cell you inverted. You may not invert the same cell twice. **Note that you cannot invert cells $(1, 1)$ and (n, n).**

Example

input

```
3
4
S010
0001
1000
111F
3
S10
101
01F
5
S0101
00000
01111
11111
0001F
```

output

```
1
3 4
2
1 2
2 1
0
```

Note

For the first test case, after inverting the cell, we get the following grid:

```
S010
0001
1001
111F
```

C. Palindromifier

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Ringo found a string s of length n in his yellow submarine. The string contains only lowercase letters from the English alphabet. As Ringo and his friends love palindromes, he would like to turn the string s into a palindrome by applying two types of operations to the string.

The first operation allows him to choose i ($2 \le i \le n-1$) and to append the substring $s_2 s_3 \ldots s_i$ ($i - 1$ characters) reversed to the front of s.

The second operation allows him to choose i ($2 \le i \le n-1$) and to append the substring $s_i s_{i+1} \ldots s_{n-1}$ ($n - i$ characters) reversed to the end of s.

Note that characters in the string in this problem are indexed from 1.

For example suppose $s =$abcdef. If he performs the first operation with $i = 3$ then he appends cb to the front of s and the result will be cbabcdef. Performing the second operation on the resulted string with $i = 5$ will yield cbabcdefedc.

Your task is to help Ringo make the entire string a palindrome by applying any of the two operations (in total) **at most 30 times**. **The length of the resulting palindrome must not exceed 10^6**

It is guaranteed that under these constraints there always is a solution. Also note you do not have to minimize neither the number of operations applied, nor the length of the resulting string, but they have to fit into the constraints.

Input

The only line contains the string S ($3 \le |s| \le 10^5$) of lowercase letters from the English alphabet.

Output

The first line should contain k ($0 \le k \le 30$) — the number of operations performed.

Each of the following k lines should describe an operation in form `L i` or `R i`. L represents the first operation, R represents the second operation, i represents the index chosen.

The length of the resulting palindrome must not exceed 10^6.

Examples

input

```
abac
```

output

```
2
R 2
R 5
```

input

```
acccc
```

output

```
2
L 4
L 2
```

input

```
hannah
```

output

```
0
```

Note

For the first example the following operations are performed:

abac $\to$ abacab $\to$ abacaba

The second sample performs the following operations: acccc $\to$ cccacccc $\to$ ccccacccc

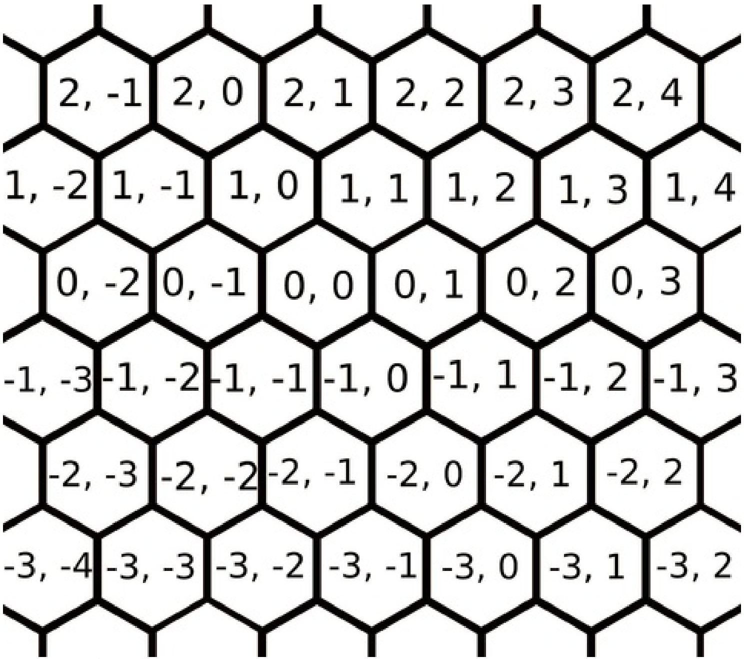
The third example is already a palindrome so no operations are required.

D. Hexagons

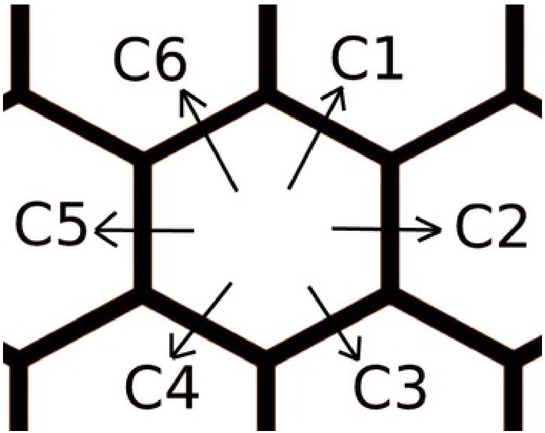
time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Lindsey Buckingham told Stevie Nicks "Go your own way". Nicks is now sad and wants to go away as quickly as possible, but she lives in a 2D hexagonal world.

Consider a hexagonal tiling of the plane as on the picture below.



Nicks wishes to go from the cell marked $(0, 0)$ to a certain cell given by the coordinates. She may go from a hexagon to any of its six neighbors you want, but there is a cost associated with each of them. The costs depend only on the direction in which you travel. Going from $(0, 0)$ to $(1, 1)$ will take the exact same cost as going from $(-2, -1)$ to $(-1, 0)$. The costs are given in the input in the order c_1, c_2, c_3, c_4, c_5, c_6 as in the picture below.



Print the smallest cost of a path from the origin which has coordinates $(0, 0)$ to the given cell.

Input

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

The first line of each test case contains two integers x and y ($-10^9 \le x, y \le 10^9$) representing the coordinates of the target hexagon.

The second line of each test case contains six integers c_1, c_2, c_3, c_4, c_5, c_6 ($1 \le c_1, c_2, c_3, c_4, c_5, c_6 \le 10^9$) representing the six costs of the making one step in a particular direction (refer to the picture above to see which edge is for each value).

Output

For each testcase output the smallest cost of a path from the origin to the given cell.

Example

input

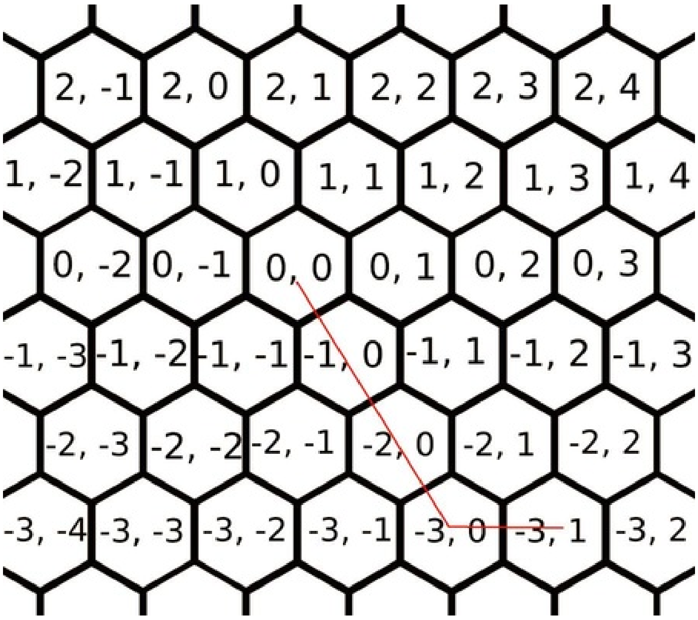
```
2
-3 1
1 3 5 7 9 11
1000000000 1000000000
1000000000 1000000000 1000000000 1000000000 1000000000 1000000000
```

output

```
18
1000000000000000000
```

Note

The picture below shows the solution for the first sample. The cost 18 is reached by taking c_3 3 times and c_2 once, amounting to $5 + 5 + 5 + 3 = 18$.



E. Swedish Heroes

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

While playing yet another strategy game, Mans has recruited n Swedish heroes, whose powers which can be represented as an array a.

Unfortunately, not all of those mighty heroes were created as capable as he wanted, so that he decided to do something about it. In order to accomplish his goal, he can pick two consecutive heroes, with powers a_i and a_{i+1}, remove them and insert a hero with power $-(a_i + a_{i+1})$ back in the same position.

For example if the array contains the elements $[5, 6, 7, 8]$, he can pick 6 and 7 and get $[5, -(6+7), 8] = [5, -13, 8]$.

After he will perform this operation $n-1$ times, Mans will end up having only one hero. He wants his power to be as big as possible. What's the largest possible power he can achieve?

Input

The first line contains a single integer n ($1 \le n \le 200000$).

The second line contains n integers $a_1, a_2, \ldots, a_n$ ($-10^9 \le a_i \le 10^9$) — powers of the heroes.

Output

Print the largest possible power he can achieve after $n-1$ operations.

Examples

input

```
4
5 6 7 8
```

output

```
26
```

input

```
5
4 -5 9 -2 1
```

output

```
15
```

Note

Suitable list of operations for the first sample:

$$[5, 6, 7, 8] \to [-11, 7, 8] \to [-11, -15] \to [26]$$