



## Task 5: Unlucky Floors

When Rar the Cat went to Taiwan for IOI 2014, he was accommodated in a hotel. During his stay, he realised that certain floors are ‘missing’ from the hotel building. Namely, he observed that numbers containing 4 and 13 as substrings are omitted from the floor numberings. This is because 4 and 13 are considered unlucky numbers and are purposely left out in the numbering. For simplicity, we will refer to this numbering scheme as the lucky numbering scheme, as it omits the unlucky numbers. The table below shows the first 20 floors in a lucky numbering scheme as well as the conventional numbering scheme.

Conventional	Lucky
1	1
2	2
3	3
4	5
5	6
6	7
7	8
8	9
9	10
10	11
11	12
12	15
13	16
14	17
15	18
16	19
17	20
18	21
19	22
20	23

However, Rar the Cat feels that such a numbering scheme is not legitimate and wants to be able to convert floors between the lucky and conventional numbering scheme. For example, floor 6 in the lucky numbering scheme will be floor 5 in the conventional numbering scheme and floor 15 will actually be floor 12. Hence, given a floor number in the lucky numbering scheme, Rar the Cat wants you to compute which floor it will be in the conventional numbering scheme and vice-versa.



## Input

Your program must read from standard input.

The input will start with a single integer,  $N$ , in a single line.  $N$  denotes how many floor numbers Rar the Cat wants you to convert for him.

$N$  lines will then follow with 2 integers each, the  $i^{th}$  line will contain  $T_i$  and  $X_i$ .

If  $T_i$  is 1, you are to convert  $X_i$  from the lucky numbering scheme to the conventional numbering scheme and print the result in a single line. However, if  $X_i$  is not a valid number in the lucky numbering scheme, print  $-1$  as the answer instead.

If  $T_i$  is 2, you are to convert  $X_i$  from the conventional numbering scheme to the lucky numbering scheme and print the result on a single line.

## Output

Your program must output to standard output only.

Output a total of  $N$  lines with 1 integer each. For each  $i$ , output the answer to  $T_i$  and  $X_i$ .

It is guaranteed that the answer will fit in a 64-bit signed integer. Refer to Sample Testcase 4 and 5 for more information.

## Subtasks

The maximum execution time on each instance is 2.5s. Your program will be tested on sets of input instances that satisfies the following restrictions:

Subtask	Marks	$N$	$T_i$	$X_i$
1	5	$0 < N \leq 50$	$T_i = 1$ or 2	$0 < X_i \leq 25$
2	12	$0 < N \leq 50$	$T_i = 1$ or 2	$0 < X_i \leq 100\,000$
3	18	$0 < N \leq 100\,000$	$T_i = 1$ or 2	$0 < X_i \leq 100\,000$
4	11	$0 < N \leq 100\,000$	$T_i = 1$	$X_i = 10^K - 1$ where $1 \leq K \leq 16$
5	37	$0 < N \leq 100\,000$	$T_i = 1$	$0 < X_i \leq 10^{16}$
6	17	$0 < N \leq 100\,000$	$T_i = 1$ or 2	$0 < X_i \leq 10^{16}$



## Sample Testcase 1

This testcase is only valid for subtasks 1, 2, 3, 5 and 6.

Input	Output
8	1
1 1	-1
1 4	12
1 15	21
1 25	1
2 1	5
2 4	18
2 15	29
2 25	

## Sample Testcase 2

This testcase is only valid for subtasks 2, 3 and 6 only.

Input	Output
10	1
1 1	5
2 4	12
1 15	18
2 15	22
1 26	-1
1 131	178
2 131	1995
2 1337	56160
1 100000	190508
2 100000	

## Sample Testcase 3

This testcase is only valid for subtasks 4, 5 and 6 only.

Input	Output
2	8
1 9	245967827040
1 9999999999999	



## Sample Testcase 4

This testcase is only valid for subtasks 5 and 6 only.

Input	Output
5	241928778399
1 987328938823	13999321852875
1 75732858587173	-1
1 4444444444444444	-1
1 1313131313131313	1534593233484559
1 100000000000000000	

## Sample Testcase 5

This testcase is only valid for subtask 6 only.

Input	Output
5	5110985302888
2 987328938823	500859079673722
2 75732858587173	30071998020860537
2 4444444444444444	8755153350232701
2 1313131313131313	76732116285952928
2 100000000000000000	