Codeforces Round 1006 (Div. 3)

A. New World, New Me, New Array

1 second, 256 megabytes

Natsume Akito has just woken up in a new world and immediately received his first quest! The system provided him with an array a of n zeros, an integer k, and an integer p.

In one operation, Akito chooses two integers i and x such that $1 \le i \le n$ and $-p \le x \le p$, and performs the assignment $a_i = x$.

Akito is still not fully accustomed to controlling his new body, so help him calculate the minimum number of operations required to make the sum of all elements in the array equal to k, or tell him that it is impossible.

Input

The first line of input contains one integer t $(1 \le t \le 1000)$ — the number of test cases.

The only line of each test case contains three integers n, k, p ($1 \le n \le 50$, $-2500 \le k \le 2500$, $1 \le p \le 50$) — the length of the array, the required sum, and the boundary of the segment from which numbers can be replaced.

Output

For each test case, output the minimum number of operations to achieve the final sum k in the array, or -1 if it is impossible to achieve the sum k.

input

```
8
21 100 10
9 -420 42
5 -7 2
13 37 7
10 0 49
1 10 9
7 -7 7
20 31 1
```

output

```
10
-1
4
6
0
-1
1
-1
```

In the fifth example, the sum of the array is initially zero, so no operations are needed.

In the sixth example, the maximum sum in the array that we can achieve is 9 (by assigning the number 9 to the single element), so the sum 10 cannot be obtained by any operations.

In the seventh example, only one operation $a_3 = -7$ is needed.

B. Having Been a Treasurer in the Past, I Help Goblins Deceive

2 seconds, 256 megabytes

After completing the first quest, Akito left the starting cave. After a while, he stumbled upon a goblin village.

Since Akito had nowhere to live, he wanted to find out the price of housing. It is well known that goblins write numbers as a string of characters '-' and '_', and the value represented by the string s is the number of distinct subsequences* of the string s that are equal to the string "-_-" (this is very similar to goblin faces).

For example, the string $s =$"-_--_-" represents the number 6, as it has 6 subsequences "-_-":

1. $s_1 + s_2 + s_3$
2. $s_1 + s_2 + s_4$
3. $s_1 + s_2 + s_6$
4. $s_1 + s_5 + s_6$
5. $s_3 + s_5 + s_6$
6. $s_4 + s_5 + s_6$

Initially, the goblins wrote a random string-number s in response to Akito's question, but then they realized that they wanted to take as much gold as possible from the traveler. To do this, they ask you to rearrange the characters in the string s so that the value of the number represented by the string s is maximized.

---

* A subsequence of a string a is a string b that can be obtained by deleting several (possibly 0) characters from a. Subsequences are considered different if they are obtained by deleting different sets of indices.

Input

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

In the first line of each test case, there is one number n ($1 \le n \le 2 \cdot 10^5$) — the length of the string written by the goblins.

In the second line of each test case, there is a string s of length n, consisting only of characters '-' and '_' — the string written by the goblins.

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

Output

For each test case, you need to output one number — the maximum number of subsequences equal to the string "-_-", if the characters in the string s are optimally rearranged.

input

```
8
3
--_
5
__-__
9
--__-_---
4
_--_
10
_-_-_-_-_-
7
_------
1
-
2
_-
```

output

```
1
0
27
2
30
9
0
0
```

In the first test case, it is beneficial to rearrange the characters to form the string "-_-". This is the only string of three characters that has at least one subsequence "-_-".

In the second test case, there is only one character "-", and at least two are needed for the subsequence "-_-". This means that for any rearrangement of characters, the answer will be 0.

In the seventh and eighth test cases, the length of the string $n < 3$, which means that subsequences of length 3 do not exist.

C. Creating Keys for StORages Has Become My Main Skill

2 seconds, 256 megabytes

Akito still has nowhere to live, and the price for a small room is everywhere. For this reason, Akito decided to get a job at a bank as a key creator for storages.

In this magical world, everything is different. For example, the key for a storage with the code (n, x) is an array a of length n such that:

- $a_1 \mid a_2 \mid a_3 \mid \ldots \mid a_n = x$, where $a \mid b$ is the bitwise "OR" of numbers a and b.
- $\mathrm{MEX}(\{a_1, a_2, a_3, \ldots, a_n\})^*$ is maximized among all such arrays.

Akito diligently performed his job for several hours, but suddenly he got a headache. Substitute for him for an hour; for the given n and x, create any key for the storage with the code (n, x).

$^*\mathrm{MEX}(S)$ is the minimum non-negative integer z such that z is not contained in the set S and all $0 \le y < z$ are contained in S.

Input

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

In the only line of each test case, two numbers n and x ($1 \le n \le 2 \cdot 10^5, 0 \le x < 2^{30}$) are given — the length of the array and the desired value of the bitwise "OR".

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

Output

For each test case, output n integers a_i ($0 \le a_i < 2^{30}$) — the elements of the key array that satisfy all the conditions.

If there are multiple suitable arrays, output any of them.

input

```
9
1 69
7 7
5 7
7 3
8 7
3 52
9 11
6 15
2 3
```

output

```
69
6 0 3 4 1 2 5
4 1 3 0 2
0 1 2 3 2 1 0
7 0 6 1 5 2 4 3
0 52 0
0 1 8 3 0 9 11 2 10
4 0 3 8 1 2
0 3
```

D. For Wizards, the Exam Is Easy, but I Couldn't Handle It

2 seconds, 256 megabytes

Akito got tired of being a simple locksmith at a bank, so he decided to enroll in the Magical Academy and become the best wizard in the world! However, to enroll, he needed to solve a single problem on the exam, which the ambitious hero could not manage.

In the problem, he was given an array a of length n. He needed to minimize the number of inversions* in the array after applying the spell **exactly once**. The spell was simple; to apply it, Akito had to choose two numbers l and r such that $1 \le l \le r \le n$ and perform a cyclic shift of the subarray from l to r one position to the left.

More formally, Akito selects the subarray $[l, r]$ and modifies the array as follows:

- From the original array $[a_1, a_2, \ldots, a_{l-1}, \mathbf{a_l}, \mathbf{a_{l+1}}, \ldots, \mathbf{a_{r-1}}, \mathbf{a_r}, a_{r+1}, \ldots, a_{n-1}, a_n]$, he obtains the array $[a_1, a_2, \ldots, a_{l-1}, \mathbf{a_{l+1}}, \mathbf{a_{l+2}}, \ldots, \mathbf{a_{r-1}}, \mathbf{a_r}, \mathbf{a_l}, a_{r+1}, \ldots, a_{n-1}, a_n$.

Akito is eager to start his studies, but he still hasn't passed the exam. Help him enroll and solve the problem!

*An inversion in an array b of length m is defined as a pair of indices (i, j) such that $1 \le i < j \le m$ and $b_i > b_j$. For example, in the array $b = [3, 1, 4, 1, 5]$, the inversions are the pairs of indices $(1, 2)$, $(1, 4)$, $(3, 4)$.

Input

The first line of input contains a number t ($1 \le t \le 10^4$) — the number of test cases.

In the first line of each test case, there is a number n ($1 \le n \le 2000$) — the length of the array a.

In the second line of each test case, there are n numbers a_i ($1 \le a_i \le 2000$) — the elements of the array a.

It is guaranteed that the sum of n^2 across all test cases does not exceed $4 \cdot 10^6$.

Output

For each test case, output two numbers l and r ($1 \le l \le r \le n$) — the boundaries of the subarray that should be chosen so that after applying the spell, the number of inversions in the array is minimized.

If there are multiple suitable pairs of boundaries, you may output any of them.

input

```
9
7
1 4 3 2 5 3 3
6
1 4 3 2 5 3
8
7 6 5 8 4 3 2 1
10
1 1 1 5 1 1 5 6 7 8
2
1337 69
4
2 1 2 1
3
998 244 353
3
1 2 1
9
1 1 2 3 5 8 13 21 34
```

output

```
2 7
2 4
1 8
4 6
1 2
1 4
1 3
2 3
5 5
```

In the first example, the array $[1, 4, 3, 2, 5, 3, 3]$ will turn into the array $[1, 3, 2, 5, 3, 3, 4]$. The inversions in it are $(2, 3)$, $(4, 5)$, $(4, 6)$ and $(4, 7)$. It can be shown that it is not possible to achieve fewer than 4 inversions.

In the second example, the array $[1, 4, 3, 2, 5, 3]$ will turn into $[1, 3, 2, 4, 5, 3]$. The inversions in it are $(2, 3)$, $(4, 6)$, and $(5, 6)$. The pair $l = 2$ and $r = 6$ also works, then the array will turn into $[1, 3, 2, 5, 3, 4]$, which also has 3 inversions — $(2, 3)$, $(4, 5)$, and $(4, 6)$. It can be shown that it is not possible to achieve fewer than 3 inversions.

In the fourth example, choosing $l = 4$ and $r = 6$ transforms the array into $[1, 1, 1, 1, 1, 5, 5, 6, 7, 8]$. It is sorted, and therefore, there are no inversions.

In the last example, the array is initially sorted, so any operation on a segment of length at least 2 will only increase the number of inversions.

E. Do You Love Your Hero and His Two-Hit Multi-Target Attacks?

3 seconds, 256 megabytes

Akito decided to study a new powerful spell. Since it possesses immeasurable strength, it certainly requires a lot of space and careful preparation. For this, Akito went out into the field. Let's represent the field as a Cartesian coordinate system.

For the spell, Akito needs to place $0 \le n \le 500$ staffs at **distinct integer coordinates** in the field such that there will be **exactly** k **pairs** (i, j) such that $1 \le i < j \le n$ and $\rho(i, j) = d(i, j)$.

Here, for two points with integer coordinates $a = (x_a, y_a)$ and $b = (x_b, y_b)$, $\rho(a, b) = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$ and $d(a, b) = |x_a - x_b| + |y_a - y_b|$.

Input

The first line of input contains a single number t $(1 \le t \le 1000)$ — the number of test cases.

In the only line of each test case, there is a single number k ($0 \le k \le 10^5$) — the number of pairs of staffs for which the equality $\rho(i, j) = d(i, j)$ must hold.

Output

For each test case, the first line of output should print the number n ($0 \le n \le 500$) — the number of placed staffs.

In the following n lines, pairs of **integers** x_i, y_i $(-10^9 \le x_i, y_i \le 10^9)$ should be printed — the coordinates of the i-th staff. The points in which staffs stand must be distinct.

input

```
3
0
2
5
```

output

```
6
69 52
4 20
789 9308706
1337 1337
-1234 -5678
23456178 707
10
-236 -346262358
273568 6435267
2365437 31441367
246574 -45642372
-236 56
4743623 -192892
10408080 -8173135
-237415357 31441367
-78125638 278
56 143231
5
1 1
2 1
1 5
3 5
1 10
```

F. Goodbye, Banker Life

3 seconds, 256 megabytes

Monsters are approaching the city, and to protect it, Akito must create a protective field around the city. As everyone knows, protective fields come in various levels. Akito has chosen the field of level n. To construct the field, a special phrase is required, which is the n-th row of the Great Magical Triangle, represented as a two-dimensional array. We will call this array T.

The triangle is defined as follows:

- In the i-th row, there are i integers.
- The single integer in the first row is k.
- Let the j-th element of the i-th row be denoted as $T_{i,j}$. Then

$$T_{i,j} = \begin{cases} T_{i-1,j-1} \oplus T_{i-1,j}, & \text{if } 1 < j < i \\ T_{i-1,j}, & \text{if } j = 1 \\ T_{i-1,j-1}, & \text{if } j = i \end{cases}$$

where $a \oplus b$ is the bitwise exclusive "OR"(XOR) of the integers a and b. Help Akito find the integers in the n-th row of the infinite triangle before the monsters reach the city.

Input

The first line contains the integer t $(1 \le t \le 10^4)$ — the number of test cases.

In the only line of each test case, there are two integers n and k ($1 \le n \le 10^6,\ 1 \le k < 2^{31}$) — the row index that Akito needs and the integer in the first row of the Great Magical Triangle, respectively.

It is guaranteed that the sum of n across all test cases does not exceed 10^6.

Output

For each test case, output n integers — the elements of the n-th row of the Great Magical Triangle.

input

```
5
1 5
2 10
3 16
9 1
1 52
```

output

```
5
10 10
16 0 16
1 0 0 0 0 0 0 0 1
52
```

In the first example, the first row of the Great Magical Triangle is $[5]$ by definition.

In the second example, $T_{2,1} = T_{1,1} = 10$ and $T_{2,2} = T_{1,1} = 10$.

G. I've Been Flipping Numbers for 300 Years and Calculated the Sum

2 seconds, 256 megabytes

After three hundred years of slime farming, Akito finally obtained the magical number n. Upon reaching the merchant, he wanted to exchange the number for gold, but the merchant gave the hero a quest.

The merchant said that for the quest, the skill $\operatorname{rev}(n, p)$ would be required, which Akito, by happy coincidence, had recently learned. $\operatorname{rev}(n, p)$ represents the following procedure:

1. Write the number n in base p, let this representation be $n = \overline{n_{\ell-1} \ldots n_1 n_0}$, where ℓ is the length of the base p representation of the number n.
2. Reverse the base p representation, let this be $m = \overline{n_0 n_1 \ldots n_{\ell-1}}$.
3. Convert the number m back to decimal and return it as the result.

The merchant's quest was to calculate the sum $x = \sum\limits_{p=2}^{k} \operatorname{rev}(n, p)$. Since this number can be quite large, only the remainder of x when divided by $10^9 + 7$ is required. The merchant also mentioned that the previous traveler had been calculating this sum for three hundred years and had not finished it. But you will help Akito finish it faster, right?

Input

The first line contains the number t ($1 \le t \le 5000$) — the number of test cases.

In the only line of each test case, two numbers n and k are given ($1 \le n \le 3 \cdot 10^5, 2 \le k \le 10^{18}$) — the magical number and the upper limit for summation.

Note that **the sum of n across all test cases is not bounded**.

Output

For each test case, you need to output a single number — the remainder of $x = \sum\limits_{p=2}^{k} \operatorname{rev}(n, p)$ when divided by $10^9 + 7$.

input

```
12
3 2
42 52
1 10
4 4
16 2
69 69
9 3
19 84
9982 44353
100000 1000000007
17 30
777 1000000000000000000
```

output

```
3
7594
9
6
1
33471
10
2006
120792461
584502117
775
46058362
```

In the third test case, $n = 1$. The number one in any numeral system is represented by a single digit, which means $\operatorname{rev}(1, p) = 1$ for any $p \ge 2$.

Thus, $x = \sum\limits_{p=2}^{k} 1 = \sum\limits_{p=2}^{10} 1 = 10 - 2 + 1 = 9$.

In the fourth test case, $x = \operatorname{rev}(4, 2) + \operatorname{rev}(4, 3) + \operatorname{rev}(4, 4)$. Let's calculate each term:

- $4 = 100_2 \to \operatorname{rev}(4, 2) = 001_2 = 1$
- $4 = 11_3 \to \operatorname{rev}(4, 3) = 11_3 = 4$
- $4 = 10_4 \to \operatorname{rev}(4, 4) = 01_4 = 1$

Thus, $x = 1 + 4 + 1 = 6$.

In the seventh test case, $x = \operatorname{rev}(9, 2) + \operatorname{rev}(9, 3)$. Let's calculate each term:

- $9 = 1001_2 \to \operatorname{rev}(9, 2) = 1001_2 = 9$
- $9 = 100_3 \to \operatorname{rev}(9, 3) = 001_3 = 1$

Thus, $x = 9 + 1 = 10$.