A. Simple Permutation

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

Given an integer n. Construct a permutation $p_1, p_2, \ldots, p_n$ of length n that satisfies the following property:

For $1 \le i \le n$, define $c_i = \lceil \frac{p_1+p_2+\ldots+p_i}{i} \rceil$, then among $c_1, c_2, \ldots, c_n$ there must be at least $\lfloor \frac{n}{3} \rfloor - 1$ prime numbers.

Input

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

In a single line of each test case, there is a single integer n $(2 \le n \le 10^5)$ — the size of the permutation.

Output

For each test case, output the permutation $p_1, p_2, \ldots, p_n$ of length n that satisfies the condition. It is guaranteed that such a permutation always exists.

Standard Input	Standard Output
3 2 3 5	2 1 2 1 3 2 1 3 4 5

Note

In the first test case, $c_1 = \lceil \frac{2}{1} \rceil = 2$, $c_2 = \lceil \frac{2+1}{2} \rceil = 2$. Both are prime numbers.

In the third test case, $c_1 = \lceil \frac{2}{1} \rceil = 2$, $c_2 = \lceil \frac{3}{2} \rceil = 2$, $c_3 = \lceil \frac{6}{3} \rceil = 2$, $c_4 = \lceil \frac{10}{4} \rceil = 3$, $c_5 = \lceil \frac{15}{5} \rceil = 3$. All these numbers are prime.

,

B1. Canteen (Easy Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

This is the easy version of the problem. The difference between the versions is that in this version, $k = 0$. You can hack only if you solved all versions of this problem.

Ecrade has two sequences $a_0, a_1, \ldots, a_{n-1}$ and $b_0, b_1, \ldots, b_{n-1}$ consisting of integers. It is guaranteed that the sum of all elements in a does not exceed the sum of all elements in b.

Initially, Ecrade can make exactly k changes to the sequence a. It is guaranteed that k does not exceed the sum of a. In each change:

- Choose an integer i ($0 \le i < n$) such that $a_i > 0$, and perform $a_i := a_i - 1$.

Then Ecrade will perform the following three operations sequentially on a and b, which constitutes one round of operations:

1. For each $0 \le i < n$: $t := \min(a_i, b_i)$, $a_i := a_i - t$, $b_i := b_i - t$;
2. For each $0 \le i < n$: $c_i := a_{(i-1) \bmod n}$;
3. For each $0 \le i < n$: $a_i := c_i$;

Ecrade wants to know the minimum number of rounds required for all elements in a to become equal to 0 after exactly k changes to a.

However, this seems a bit complicated, so please help him!

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2 \cdot 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n, k ($1 \le n \le 2 \cdot 10^5$, $k = 0$).

The second line of each test case contains n integers $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le 10^9$).

The third line of each test case contains n integers $b_0, b_1, \ldots, b_{n-1}$ ($1 \le b_i \le 10^9$).

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$. It is also guaranteed that in each test case the sum of a does not exceed the sum of b, and that k does not exceed the sum of a.

Output

For each test case, output the minimum number of rounds required for all elements in a to become equal to 0 after exactly k changes to a.

Standard Input	Standard Output
4 3 0 1 1 4 5 1 4 4 0	1 4 4 8

1 2 3 4 4 3 2 1 4 0 2 1 1 2 1 2 2 1 8 0 1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1	

Note

In this version, Ecrade cannot make changes to a.

In the first test case:

- After the first round, $a = [0, 0, 0], b = [4, 0, 0]$.

In the second test case:

- After the first round, $a = [3, 0, 0, 1], b = [3, 1, 0, 0]$;
- After the second round, $a = [1, 0, 0, 0], b = [0, 1, 0, 0]$;
- After the third round, $a = [0, 1, 0, 0], b = [0, 1, 0, 0]$;
- After the fourth round, $a = [0, 0, 0, 0], b = [0, 0, 0, 0]$.

,

B2. Canteen (Hard Version)

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   512 megabytes
```

This is the hard version of the problem. The difference between the versions is that in this version, there are no additional limits on k. You can hack only if you solved all versions of this problem.

Ecrade has two sequences $a_0, a_1, \ldots, a_{n-1}$ and $b_0, b_1, \ldots, b_{n-1}$ consisting of integers. It is guaranteed that the sum of all elements in a does not exceed the sum of all elements in b.

Initially, Ecrade can make exactly k changes to the sequence a. It is guaranteed that k does not exceed the sum of a. In each change:

- Choose an integer i ($0 \le i < n$) such that $a_i > 0$, and perform $a_i := a_i - 1$.

Then Ecrade will perform the following three operations sequentially on a and b, which constitutes one round of operations:

1. For each $0 \le i < n$: $t := \min(a_i, b_i), a_i := a_i - t, b_i := b_i - t$;
2. For each $0 \le i < n$: $c_i := a_{(i-1) \bmod n}$;
3. For each $0 \le i < n$: $a_i := c_i$;

Ecrade wants to know the minimum number of rounds required for all elements in a to become equal to 0 after exactly k changes to a.

However, this seems a bit complicated, so please help him!

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 2 \cdot 10^4$). The description of the test cases follows.

The first line of each test case contains two integers n, k ($1 \le n \le 2 \cdot 10^5$, $0 \le k \le 2 \cdot 10^{14}$).

The second line of each test case contains n integers $a_0, a_1, \ldots, a_{n-1}$ ($1 \le a_i \le 10^9$).

The third line of each test case contains n integers $b_0, b_1, \ldots, b_{n-1}$ ($1 \le b_i \le 10^9$).

It is guaranteed that the sum of n across all test cases does not exceed $2 \cdot 10^5$. It is also guaranteed that in each test case the sum of a does not exceed the sum of b, and that k does not exceed the sum of a.

Output

For each test case, output the minimum number of rounds required for all elements in a to become equal to 0 after exactly k changes to a.

Standard Input	Standard Output
8 3 0 1 1 4 5 1 4 4 0	1 4 4 8 0

1 2 3 4 4 3 2 1 4 0 2 1 1 2 1 2 2 1 8 0 1 2 3 4 5 6 7 8 8 7 6 5 4 3 2 1 3 6 1 1 4 5 1 4 4 1 1 2 3 4 4 3 2 1 4 1 2 1 1 2 1 2 2 1 4 2 2 1 1 2 1 2 2 1	2 2 1

Note

In the fifth test case, all elements in a become 0 after exactly 6 changes.

In the sixth test case, Ecrade can do exactly one change to a_3, then a will become $[1, 2, 2, 4]$.

- After the first round, $a = [3, 0, 0, 0], b = [3, 1, 0, 0]$;
- After the second round, $a = [0, 0, 0, 0], b = [0, 1, 0, 0]$.

In the seventh test case, Ecrade can do exactly one change to a_4, then a will become $[2, 1, 1, 1]$.

- After the first round, $a = [0, 1, 0, 0], b = [0, 1, 1, 0]$;
- After the second round, $a = [0, 0, 0, 0], b = [0, 0, 1, 0]$.

,

C1. Key of Like (Easy Version)

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

This is the easy version of the problem. The difference between the versions is that in this version, it is guaranteed that $k = 0$. You can hack only if you solved all versions of this problem.

A toy box is a refrigerator filled with childhood delight. Like weakness, struggle, hope ... When such a sleeper is reawakened, what kind of surprises will be waiting?

M received her toy box as a birthday present from her mother. A jewellery designer would definitely spare no effort in decorating yet another priceless masterpiece as a starry firmament with exquisitely shaped gemstones. In addition, l distinct locks secure the tiny universe of her lovely daughter: a hair clip featuring a flower design, a weathered feather pen, a balloon shaped like the letter M ... each piece obscures a precious moment.

A few days ago, M rediscovered her toy box when she was reorganizing her bedroom, along with a ring of keys uniquely designed for the toy box. Attached to the key ring are $(l + k)$ keys, of which l keys are able to open one of the l locks correspondingly, while the other k keys are nothing but counterfeits to discourage brute-force attack. To remind the correspondence, M's mother adorned each key with a gemstone of a different type. However, passing days have faded M's memory away.

"... So I have to turn to you all," M said while laying that ring of keys on the table.

K picked up the keys and examined them carefully. "The appearance of these keys unveils nothing fruitful. Thus, I am afraid that we shall inspect them sequentially."

Although everyone is willing to help M, nobody has a plan. Observing others' reactions, T suggested, "Let's play a game. Everyone tries a key in turn, and who opens the most locks is *amazing*."

n members, including M herself, take turns to unlock the toy box recursively in the same order until all the l locks are unlocked. At each turn, the current member only selects a single key and tests it on exactly one of the locks. To open the toy box as soon as possible, every member chooses the key and the lock that maximize the probability of being a successful match. If there are multiple such pairs, a member will randomly choose one of such pairs with equal probability. Apparently, if a lock has been matched with a key, then neither the lock nor the key will be chosen again in following attempts.

Assume that at the very beginning, the probability that a lock can be opened by any key is equal. If everyone always tries the optimal pairs of keys and locks based on all the historical trials, what will the expected number of successful matches be for each member?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The only line of the input contains three integers, n, l, k ($1 \le n \le 100, 1 \le l \le 5000, k = 0$) — the number of members participating in the game, the number of locks, and the number of counterfeit keys.

It is guaranteed that the sum of l across all test cases does not exceed 5000.

Output

For each test case, output a single line with n integers $e_1, \ldots, e_n$, where e_i represents the expected number of successful matches, modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the exact answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer e_i that $0 \le x < M$ and $e_i \cdot q \equiv p \pmod{M}$.

Standard Input	Standard Output
4 3 1 0 3 2 0 2 5 0 9 104 0	1 0 0 500000004 1 500000004 200000004 800000008 869203933 991076635 39374313 496894434 9358446 51822059 979588764 523836809 38844739

Note

For the first test case, there is only 1 lock, so the first member opens the only lock with the only key undoubtedly.

For the second test case, there are exactly 2 locks and 2 keys, with each key corresponding to one of the locks. Without extra information, the first member randomly chooses a key and a lock with equal probabilities, for which the probability of success is $1/2$.

- If the first member succeeds, the second member will open the other lock with the other key.
- If the first member fails, then the key she selected can open the other lock, and the other key must correspond to the lock she chose. This information allows both the second and the third member to open a lock.

In conclusion, the expected numbers of successful matches will be:

$$
\begin{aligned}
e_1 &= \frac{1}{2} \times 1 + \frac{1}{2} \times 0 = \frac{1}{2} \equiv 500,000,004 \pmod{10^9+7}, \\
e_2 &= \frac{1}{2} \times 1 + \frac{1}{2} \times 1 = 1, \\
e_3 &= \frac{1}{2} \times 0 + \frac{1}{2} \times 1 = \frac{1}{2} \equiv 500,000,004 \pmod{10^9+7}.
\end{aligned}
$$

,

C2. Key of Like (Hard Version)

Input file: standard input
Output file: standard output
Time limit: 3 seconds
Memory limit: 512 megabytes

This is the hard version of the problem. The difference between the versions is that in this version, k can be non-zero. You can hack only if you solved all versions of this problem.

A toy box is a refrigerator filled with childhood delight. Like weakness, struggle, hope ... When such a sleeper is reawakened, what kind of surprises will be waiting?

M received her toy box as a birthday present from her mother. A jewellery designer would definitely spare no effort in decorating yet another priceless masterpiece as a starry firmament with exquisitely shaped gemstones. In addition, l distinct locks secure the tiny universe of her lovely daughter: a hair clip featuring a flower design, a weathered feather pen, a balloon shaped like the letter M ... each piece obscures a precious moment.

A few days ago, M rediscovered her toy box when she was reorganizing her bedroom, along with a ring of keys uniquely designed for the toy box. Attached to the key ring are $(l+k)$ keys, of which l keys are able to open one of the l locks correspondingly, while the other k keys are nothing but counterfeits to discourage brute-force attack. To remind the correspondence, M's mother adorned each key with a gemstone of a different type. However, passing days have faded M's memory away.

"... So I have to turn to you all," M said while laying that ring of keys on the table.

K picked up the keys and examined them carefully. "The appearance of these keys unveils nothing fruitful. Thus, I am afraid that we shall inspect them sequentially."

Although everyone is willing to help M, nobody has a plan. Observing others' reactions, T suggested, "Let's play a game. Everyone tries a key in turn, and who opens the most locks is *amazing*."

n members, including M herself, take turns to unlock the toy box recursively in the same order until all the l locks are unlocked. At each turn, the current member only selects a single key and tests it on exactly one of the locks. To open the toy box as soon as possible, every member chooses the key and the lock that maximize the probability of being a successful match. If there are multiple such pairs, a member will randomly choose one of such pairs with equal probability. Apparently, if a lock has been matched with a key, then neither the lock nor the key will be chosen again in following attempts.

Assume that at the very beginning, the probability that a lock can be opened by any key is equal. If everyone always tries the optimal pairs of keys and locks based on all the historical trials, what will the expected number of successful matches be for each member?

Input

Each test contains multiple test cases. The first line contains the number of test cases t ($1 \le t \le 100$). The description of the test cases follows.

The only line of the input contains three integers, n, l, k ($1 \le n \le 100, 1 \le l \le 5000, 0 \le k \le 25$) — the number of members participating in the game, the number of locks, and the number of counterfeit keys.

It is guaranteed that the sum of l across all test cases does not exceed 5000.

Output

For each test case, output a single line with n integers $e_1, \ldots, e_n$, where e_i represents the expected number of successful matches, modulo $10^9 + 7$.

Formally, let $M = 10^9 + 7$. It can be shown that the exact answer can be expressed as an irreducible fraction $\frac{p}{q}$, where p and q are integers and $q \not\equiv 0 \pmod{M}$. Output the integer equal to $p \cdot q^{-1} \bmod M$. In other words, output such an integer e_i that $0 \le x < M$ and $e_i \cdot q \equiv p \pmod{M}$.

Standard Input	Standard Output
4 3 1 4 3 2 0 25 2 5 4 102 9	800000006 800000006 400000003 500000004 1 500000004 142857144 166666668 615646263 639455787 234126986 257936510 195918369 502040820 478316330 81264173 190523433 471438023 23809524 0 0 0 0 0 0 0 0 0 0 0 0 568832210 85779764 969938175 375449967

Note

For the first test case, there is only 1 lock, so the strategy will always be choosing any key that no one has ever tried. Since there are $1 + 4 = 5$ keys in total, the probability that each member successfully opens the lock will be $2/5, 2/5, 1/5$ respectively, which are also the expected numbers of successful matches.

For the second test case, there are exactly 2 locks and 2 keys, with each key corresponding to one of the locks. Without extra information, the first member randomly chooses a key and a lock with equal probabilities, for which the probability of success is $1/2$.

- If the first member succeeds, the second member will open the other lock with the other key.
- If the first member fails, then the key she selected can open the other lock, and the other key must correspond to the lock she chose. This information allows both the second and the third member to open a lock.

In conclusion, the expected numbers of successful matches will be:

$$e_1 = \frac{1}{2} \times 1 + \frac{1}{2} \times 0 = \frac{1}{2} \equiv 500,000,004 \pmod{10^9 + 7},$$
$$e_2 = \frac{1}{2} \times 1 + \frac{1}{2} \times 1 = 1,$$
$$e_3 = \frac{1}{2} \times 0 + \frac{1}{2} \times 1 = \frac{1}{2} \equiv 500,000,004 \pmod{10^9 + 7}.$$

,

D. Conditional Operators

```
Input file:     standard input
Output file:    standard output
Time limit:     2 seconds
Memory limit:   256 megabytes
```

In C++, the conditional operator `?:` is used as the value of `x?y:z` is y if x is true; otherwise, the value is z. x, y, and z may also be expressions. It is right-associated; that is, `a?b:c?d:e` is equivalent to `a?b:(c?d:e)`. 0 means false and 1 means true.

Given a binary string with length $2n+1$, you need to show whether the value of the expression can be 1 after inserting n conditional operators into the string. You can use parentheses. For example, the string `10101` can be transformed into `(1?0:1)?0:1`, whose value is 1.

Input

The first line contains a single integer t $(1 \le t \le 10\,000)$, the number of test cases. The description of the test cases follows.

In the first line of each test case, there is a single integer n $(1 \le n \le 1.5 \cdot 10^5)$.

In the second line of each test case, there is a binary string of length $2n+1$.

It is guaranteed that the sum of n across all test cases does not exceed $1.5 \cdot 10^5$.

Output

For each test case, on the first line, output `Yes` if the string can be transformed into an expression of value 1; otherwise, output `No`.

If the answer is `Yes`, output the expression on the second line. You can use parentheses, but the order of the characters in the original string must remain the same. The length of your expression must be no more than $10n + 1000$.

Standard Input	Standard Output
2 2 10101 2 00000	Yes (1?0:1)?(0):1 No

Note

The first test case — is the one mentioned in the problem description.

In the second test case, it is clear that regardless of how the conditional operator is used, the result will always be zero.

,

E. Black Cat Collapse

```
Input file:     standard input
Output file:    standard output
Time limit:     3 seconds
Memory limit:   1024 megabytes
```

The world of the black cat is collapsing.

In this world, which can be represented as a rooted tree with root at node 1, Liki and Sasami need to uncover the truth about the world.

Each day, they can explore a node u that has not yet collapsed. After this exploration, the black cat causes u and all nodes in its subtree to collapse. Additionally, at the end of the i th day, if it exists, the number $n - i + 1$ node will also collapse.

For each i from 1 to n, determine the number of exploration schemes where Liki and Sasami explore exactly i days (i.e., they perform exactly i operations), with the last exploration being at node 1. The result should be computed modulo $998\,244\,353$.

Note: It is guaranteed that nodes 1 to n can form a "DFS" order of the tree, meaning there exists a depth-first search traversal where the i th visited node is i.

Input

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

The first line of each test case contains exactly one number n ($3 \le n \le 80$).

Each of the following $n - 1$ lines contains two integers u_i and v_i, representing two vertices connected by an edge ($1 \le u_i, v_i \le n$). It is guaranteed that the given edges form a tree. It is also guaranteed that the vertices can form a "DFS" traversal order.

It is guaranteed that the sum of n for all test cases does not exceed 80

Output

For each test case, print n integers, where the i th integer represents the number of exploration schemes for exactly i days, modulo $998\,244\,353$.

Standard Input	Standard Output

2 4 1 2 2 3 2 4 7 4 2 6 1 5 1 7 6 2 3 1 2	1 3 3 1 1 6 23 48 43 17 1

Note

For the first test case, the following operation sequences are legal:

$$\{1\}, \{2,1\}, \{3,1\}, \{4,1\}, \{3,2,1\}, \{4,2,1\}, \{4,3,1\}, \{4,3,2,1\}.$$