Problem A - Light

Description

You are standing in front of a row of lights. Some of these lights are on and some are off. Since these lights look strange, you want to turn them off. However, there are some problems with the circuit for controlling the lights. You can only change the state of k consecutive lights in one operation (changing the state means that if the light was originally on, it will turn off, and vice versa). You now want to know that you can turn off these lights at least several times. If you can't turn them off, output "so hard" (no need to output "")

Input format

The first line contains a string containing only 0 or 1 on the first line to represent the current switch status of the light, 0 represents off and 1 represents on, and the length of the string is less than or equal to 1000000.

the second line contains a positive integer $k(k \leq 1000000)$ in the second line, and refer to the title description for its meaning

Output format

Output a line representing your answer or "so hard"

Sample Input

```
01010
2
```

Sample Output

```
2
```

Problem B - Slogans

Description

A good company needs corporate culture , so slogans are very necessary.Therefore the company's executives camp up with their own slogan, respectively, $s_{1...n}$.

Certainly,there are some difference between slogans , each slogan s_i has a cool value v_i . Specifically , if there exists $a < b < c$, $s_{ia} = s_{ic} =' q'$, $s_{ib} =' a'$, v_i will add 1.

Now tell you all their slogans,you need to output all v_i.

Input format

The first line of the input contains a single integer $n(1 \leq n \leq 1000\ 000)$ - the number of executives.

The next N lines contains a string which is in lowercase - the slogan of each executives.

It is guaranteed the sum of length of string does not exceed $1,000,000$.

Output fotmat

Output N lines , each line contain a integer - v_i.

Sample Input

```
3
qaq
qaaq
qqaaqq
```

Sample Output

```
1
2
8
```

Problem C - An Interesting Game

Description

Mr. Feng and his girlfriend are playing a game. Each of them has countless balls. The weight of Mr. Feng's ball is x, and the weight of his girlfriend's ball is y, where x and y are co-prime(Which means $gcd(x, y) = 1$). There is a plate in the middle (the plate has no weight). They can play any round (each round is independent), and each person can choose to throw any ball (which can be zero) into it. They want to know what is the maximum integer weight that can't be obtained in the plate.

For example, when x = 2 and y = 3, it can be found that the maximum weight is 1, and it can be proved that for any k>1,there exists a,b satisfied $k = ax + by(a >= 0, b >= 0)$.

After a short game, the clever two people found that the answer to this game is $x * y - x - y$.

The next day, in order to challenge himself, Mr. Feng decided to strengthen the difficulty of the game. They want to know what is the value of the fourth largest integer weight that can't be obtained in the plate.Data guarantee this value > 0.

For example,for $x = 5, y = 7$,the answer is 13, for $1, 2, 3, 4, 6, 8, 9, 11, 13, 16, 18, 23$ are what can’t be obtained.

Input format

Each test contains multiple test cases. The first line contains the number of test cases T($1 <= T <= 100$) ,Description of the test cases follows.

The single line of each test case contains two integers x and y ($1 <= x, y <= 2e9$).

It is guaranteed that $gcd(x, y) = 1$ and the answer >0 .

Output format

For each test case, print one integer.

Sample Input

```
3
5 7
9 11
3 233
```

Sample Output

```
13
61
454
```

Problem D - Sum of Sets

Description

You have n sets of positive integers numbered from 1 to n (Obviously, a set can't contain a integer many times) . These sets are initially empty.

An empty set makes no sense, so you want to perform m operations to add elements to these sets.

In an operation $L\ R\ x$, you will add element x to every set i which $L \leq i \leq R$. If x already exists in a set, it will **NOT** be added to this set.

After m operations, you want to know the sum of every set. Note that the sum of an empty set is 0 .

Input format

The first line contains two positive integers n, m ($1 \leq n, m \leq 10^5$) .

In the next m lines, every line contains three positive integers L, R, x ($1 \leq L \leq R \leq n, 1 \leq x \leq 2 \times 10^4$) — an operation.

Output format

The output contains one line. For each i $(1 \le i \le n)$, output the sum of the set i .

Sample Input

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

Sample Output

```
3 3 3 3 2
```

Problem E - Magic Words

Description

V4yne, a famous magician, is challenged by another magician GieGie one day: GieGie summons many monsters to attack V4yne, while V4yne would fail the battle if he can not kill all of these monsters.

The monsters summoned by GieGie are all monsters of elements. Elements include fire element, wind element, and many nature elements. Each monster has only one element and each element is referred by a **type_id**.

V4yne can use many kinds of magic words. Each magic words include many elements, and each element is also referred by a **type_id**. It means, every magic words contain many values of **type_id**. Monster will be attacked by the magic words which contains the same **type_id** of the monster, and every time it will lose **1HP**. (Every time V4yne uses one magic words, he can only attack one monster.)

But V4yne's magic is not powerful, which means he can only use every magic words for limited times. And every monster has its own HP number. If a monster has **x** HP, it should be attacked **x** times, then it will die.

V4yne will win only if he can kill all the monsters. Can you make a calculation that can V4yne defeat GieGie? If he can't, can you figure out how many HP loss V4yne can make to all the monsters at most?

Input format

The first line contains integer **n**(1<=n<=300) and **m**(1<=m<=500), **n** is the number of magic words, **m** is the number of monsters.

The second line contains **n** numbers, the i-th number indicates the maximum number of times the i-th magic words can be used.

The third line contains **m** numbers, the i-th number indicates the HP of the i-th monster.

Next there are **n** lines. The first number **x** in the i-th line indicates there are **x** kinds of **type_id** contained in the i-th magic words. The next **x** different **type_id** values indicate all the elements contained in this magic words.

The following line contains **m** numbers， and the i-th number is the **type_id** of the i-th monster‘s element.

The number of every **type_id** is not bigger than the number of **n+m**.

(All datas given in the input are less than 500.)

Output fotmat

Output “**V4yne**” if V4yne can win.

Otherwise, output the **maximum damage value** that V4yne can cause to all the monsters summoned by GieGie.

(The damage value refers to the total HP reduced by all the monsters, not the number of deaths of the monsters.)

Sample Input 1

```
2 3
3 7
2 3 5
2 1 3
2 2 3
1 2 3
```

Sample Output 1

```
V4yne
```

Sample Input 2

```
3 1
100 100 100
5
1 1
1 2
1 3
4
```

Sample Output 2

```
0
```

Problem F - TangTang's Question

Description

A q days vacation is coming! Tang Tang is a sensible child. He will help his mother to buy products in the supermarket every day. There are n products on the container of the supermarket, and the type of each products is a_i. Every day, the mother would let Tang Tang buy all the $[a_{l_i}, a_{l_i+1} .. a_{r_i}]$ products home. Supermarket pricing strategy is very strange, for each type of products, the price is the distance between the first and last occurrences . Tang Tang's math skills are very poor. He hopes you can help him figure out how much money he needs to bring every day.

Input format

The first line contains an integer $n(1 \le n \le 10^5)$, indicating the number of products in the supermarket.

The second line contains n integers $a_1, a_2, \ldots, a_n (1 \le a_i \le 10^5)$

The third line contains an integer $q(1 \le q \le 10^5)$, indicating the number of query.

The i-th of the next q lines contains two integers $l'_i, r'_i (0 \le l'_i, r'_i \le 10^{18})$

$l_i = Min((l'_i \oplus lastans) mod\ n\ + 1, (r'_I \oplus lastans) mod\ n\ + 1)$

$r_i = Max((l'_i \oplus lastans) mod\ n\ + 1, (r'_i \oplus lastans) mod\ n\ + 1)$

$\oplus$ denotes **bitwise exclusive OR** operation.

$lastans$ denotes the answer to last query.

In the beginning, $lastans = 0$.

Output format

For each query, print its answer on a new line.

Sample Input

```
6
1 2 3 2 3 2
2
1 3
3 6
```

Sample output

```
2
4
```

Problem G - Play on the Graph

Description

Eva loves wandering about but she doesn't like any limitations.

So nice Colin built an **Undirected Complete Graph** containing N nodes, numbering from 1 to N. And then he presented this graph to Eva.

Then they begin to play on this graph. At the beginning, they **both** stand at the Node 1. Then, Eva is going to walk randomly on the graph while Colin stays at the Node 1 looking at her. Specifically, every second Eva will **randomly choose one edge related to the very point she stands now** to move.

The thing is that at some seconds Colin will expect Eva to be by his side, but he doesn't want to tell Eva. So he comes to ask you what's the probability of the event that Eva is also standing at Node 1 (where Colin stays) at these specific seconds.

P.S. To avoid floating point errors, Colin wants the probability module 998244353.

Input format

One line with two integers n and q $(2 \le n \le 10^{18}, 1 \le q \le 10^5)$, represents the number of the nodes in the graph, and the number of queries.

Then q lines follow, each with an integer t $(0 \le t \le 10^{18})$, represents querying the probability of the event that Eva is standing at Node 1 (where Colin stays) at the second t.

Output format

For each query t, output one line containing one integer, represents the probability of the event that Eva is standing at Node 1 (where Colin stays) at the second t.

It can be proved that the answer can be represented as an irreducible fraction $\frac{x}{y}$, and we guarantee that it can be represented while module 998244353 .

Sample Input

```
3 2
0
2
```

Sample Output

```
1
499122177
```

Problem H - Escape From The Secret Room

Description:

GieGie is trapped in a secret room, the secret room has only one door which is controlled by a circuit. GieGie has got the circuit design chart: the circuit's connection forms a tree, and there are n nodes in the circuit numbered from 1 to n, the root is always be the node 1. Except of the leaf node which is connected with controller, each of other nodes is a logic gate including AND gate, OR gate and XOR gate. The controller of the leaf node has only two signal states(0 and 1) which can be switched by GieGie. Obviously, different combination of the initial signal states may make the root node output different state of signal, the root node of the circuit is connected with the door, only when the output signal is 1 the door will be opened.
GieGie has no time to find out a correct configuration to make the ouput of the root node be 1 cause he wants to meet JieJie as soon as possible, so he plans to set the state of controllers randomly.
You should tell the expected times GieGie should set, so the door will open.

Note: A logic gate take several input signals and give an output signal which depends on the gate's type and input signals' state.
Here we assume the inputs of the some logic gate are $s_1, s_2, \cdots, s_m$
AND gate: output $s_1 \& s_2 \& \cdots \& s_m$
OR gate: output $s_1 | s_2 | \cdots | s_m$
XOR gate: output $s_1 \oplus s_2 \oplus \cdots \oplus s_m$, $\oplus$ means XOR operator.
If some gate gets only one input signal, then it should output signal the same as the input.
You can assume that the signal is always transmitting from the leaf nodes to the root node, and you can assume that only when all input signals get to the gate then it can output the signal to the next gate.

Input format

The first line contains one integer $n (1 \le n \le 10^5)$ — the number of nodes in circuit.
The next $n-1$ lines describe the circuit. These lines contain two integers u_i and v_i $(1 \le u_i, v_i \le n)$denoting a connection between u_i and v_i.
The last line contains n integers t_1 to t_n, each describe the type of the i-th logic gate or controller.$t_i = -1$ means it's a controller, $t_i = 0$ means it's a AND gate, $t_i = 1$ means it's a OR gate, $t_i = 2$ means it's a XOR gate. Only leaf nodes will be -1.

Output fotmat

It can be shown that the expected number of times GieGie switched at the end can be represented as an irreducible fraction $\frac{x}{y}$. Print this fraction modulo 998244353, i. e. value $x \cdot y^{-1} \mod 998244353$ where y^{-1} is such number that $y \cdot y^{-1} \mod 998244353 = 1$.

Sample Input

```
7
1 2
1 3
2 4
2 5
3 6
3 7
2 1 0 -1 -1 -1 -1
```

Sample Output

```
798595484
```

Hint: the answer is $\frac{8}{5}$

Problem I - GieGie and Cube

Description

Recently, GieGie is working on a easy problem.

For numbers from L to R, GieGie needs to calculate the third power of the sum of the third power of each number.(If you can't understand, read the following.)
GieGie thinks the following function can solve the problem.

```
int calc( int L , int R )
{
    int sum = 0;
    for ( int i = L ; i <= R ; ++i )
        sum += i * i * i;
    return sum * sum * sum;
}
```

But you're very thoughtful. Obviously, this function can't solve the problem when (r-l) is large. You need to help GieGie calculate correctly!

For each given integers L and R, you should print answer%12345678910.

Input format

The first line contains an integer t $(1 \le t \le 10^4)$ — the number of test cases. The description of the test cases follows.
The single line of each test case contains two integers l,r$(1 \le l \le r \le 10^9)$ — the range of the numbers.

Output fotmat

For each test case print a single integer —— the answers modulo 12345678910.

Sample Input

```
2
2 2
1000000000 1000000000
```

Sample Output

```
512
2359345890
```

Problem J - Push-ups Triggered by a Basketball

Description

Miss Congee and Mr. Zhang met at the stadium this day. Mr. Zhang uttered wild words and provoked, "Dare you come and shoot basketball?" "Why not?" Miss Congee responded.

The war is about to start. And the referee is, you!

The two take turns to shoot, only one shot at a time. Miss Congee shoots first.

As the referee, you must record a number "Count". When the game starts, it is 0.

When a person makes a successful shot, you add 1 to the "Count".

And when a person fails to shoot, he must accept the punishment: do push-ups for "Count" times, and then you clear the "Count" to zero.

In order to prevent them from wasting extra energy in the game, you decide to impose a penalty at the end of the game.

Both sides shot N times, and you recorded everyone's shots.

Now that the game is over, the next step is to calculate the number of push-ups that both sides need to do.
But Mr. Zhang does not know that you are actually a suitor of Miss Congee, so you want Miss Congee to do as few push-ups as possible, and on this basis, you want Mr. Zhang to do as many push-ups as possible.

So you are ready to tamper with the record of your shots. But too many changes are easy to be found, so you can only change one place at most, that is, change a successful one to a failure, or change a failed one to a success. Of course, you can do nothing.

Now you want to know the number of push-ups to be done by both in the optimal situation.

Input Format

A positive integer N$(1 \leq N \leq 2000)$ in the first line indicates the total number of rounds.
The second line is a 01 string of length N*2, 1 means success, 0 means failure.

Output Format

Output two integers on one line to represent the number of push-ups to be done by Miss Congee and Mr. Zhang respectively.

Sample Input

```
4
11111111
```

Sample Output

```
0 7
```

Problem K - Quaternions

Description

GieGie likes to think about some strange questions.

Now GieGie has an array a consisting of n integers.He now wants to know how many quaternions (i, j, k, w) satisfy $i < j < k < w$ and $a_i \times a_j \times a_k \times a_w$ is a square number .

The definition of a square number is a number that can be expressed as the product of two identical integers. For example, 1, 4, 9, 16, 25, and 36 are square numbers, but 2, 3, and 6 are not.

Input format

The first line of each test case contains a single integer $n(1 \leq n \leq 6000)$ ¡ª the length of the array a.

The second line of each test case contains n integers $a_1, a_2, \cdots, a_n (1 \leq a_i \leq 6000)$ ¡ª the elements of the array a.

Output format

For each test case, print one integer: the number of quaternions that meet the condition.

Sample Input

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

Sample Output

```
2
```