Final Assignment

A. And Construction

3 seconds, 256 megabytes

After mastering bitwise operations, you and I are no longer just programmers — we've become half hackers, half cryptographers.

We can see numbers as streams of bits, hidden keys to secret codes. Every AND, OR, and XOR is like a move in a secret dance to hide or reveal messages.

So,, I give you an array A of N integers, each holding a secret. But we can't just leave them lying around; we have to encode this data into another array B of size $N + 1$ such that $A_i = B_i \ \& \ B_{i+1}$ where $\&$ is the bitwise And operation.

But there is a problem not every array can be encoded this way, So if it is possible print array B or else print -1

Input

Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 10^5)$. The description of the test cases follows.

The first line of each test case contains an integer N $(1 \le N \le 10^5)$ — the length of A.

The following line contains N space-separated integers $A_1, A_2, A_3, \ldots\ldots, A_N$ $(1 \le A_i \le 10^9)$- the elements of array A

It is guaranteed that the sum of N over all test cases does not exceed $3 \cdot 10^5$.

Output

If it is possible to encode the numbers then print those encoded $N + 1$ numbers or print -1 .

input

```
5
1
1
3
4 0 4
3
1 2 3
2
3 6
5
64 4 14 8526 320
```

output

```
1 1
-1
-1
3 7 6
64 68 14 8526 8526 320
```

B. Binary Search Blunder

2 s., 256 MB

Ohh!!! Binary search was taught so long ago... and we all know that binary search works only on sorted arrays.But one of our dear friends, X, seems to have forgotten that crucial detail!

For the final assignment I give him a permutation A of size N, a number x, and ask him to find it in the array.

Poor fellow X, full of confidence, writes his binary search code without sorting:

```
int lo = 1, hi = n;
int ans = -1;
while (lo <= hi) {
    int mid = (lo + hi) / 2;
    if (v[mid] == x) {
         ans = x;
         break;
    }
    if (v[mid] < x) lo = mid + 1;
    else hi = mid - 1;
}
```

```
cout<<ans;
```

He happily runs it, completely unaware that binary search works only on sorted arrays!
But wait! You are his best friend!

You don't want to embarrass him. So you come up with a genius idea to help him. You decide to change some numbers in the array, so that his code outputs x correctly instead of -1 even though the array is not sorted.

But no one should get doubt on you... So you can only do the following operations:

- Select any i $(1 \le i \le n)$ and put $A_i = k$, where $(0 \le k \le 10^9)$ and $\mathbf{k} \neq \mathbf{x}$
- The number of times you do the above operation should be as minimum as possible

So , find the minimum number of times you can perform the above operation and save your friend and also escape from my suspicion..

Input
Each test contains multiple test cases. The first line contains the number of test cases t $(1 \le t \le 5.10^4)$. The description of the test cases follows.

The first line of each test case contains integers N, x $(1 \le x \le N \le 2 \cdot 10^5)$ — the length of A and the element to find.

The following line contains N space-separated integers $A_1, A_2, A_3, \ldots\ldots, A_N$ $(1 \le A_i \le N)$ - the elements of array A. It is guaranteed that A is a permutation (i.e); every number from 1 to N appears only once in the array .

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

Output
Print the minimum number of changes you do to the array

input
```
2
5 1
1 2 3 4 5
5 1
5 4 3 2 1
```
output
```
0
2
```

For the first case, the array is sorted so no need of any changes.

For the second case, Change the array as [5, 4, 0, 0, 1] so your friend's binary search code correctly finds 1 . So the minimum number of changes made is 2 . You cannot make less than 2 changes and achieve the answer .

C. Combinatorial Love

3 s, 256 megabytes

Wow... It's Valentines day and I have to send letters and gifts to all my N girlfriends. Girlfriend i needs to receive gift i

But wait , maintaining N girlfriends is not a easy task... So sometimes, by mistake the gifts may get shuffled and if girlfriend i gets the gift j where $i \neq j$, then she breaks up with me.... Noooo..... This should not happen....

I don't want to become a single after gift distribution... So what is the probability of me having atleast 1 girlfriend after valentines day...

Print the answer modulo $10^9 + 7$.

Input
The first line of input contains an integer t $(1 \le t \le 10^6)$ - the number of test cases.

Each test case contains a integer N $(1 \le N \le 10^6)$ - number of girlfriends I have.

Output
Output the probability of me having at least 1 girlfriend after giving gifts modulo $10^9 + 7$.

input
```
3
1
2
3
```
output
```
1
500000004
666666672
```

```
666666672
```

In the first case, I have only 1 girlfriend so there is no possibility of gift mismatch and I will surely have at least 1 girlfriend after giving gift. So the Probability is 1 .

In the second case , if the gifts were in order [1, 2] then I will have at least 1 girlfriend and the Total number of possibilities of gift distribution are [1, 2] and [2, 1] . So probability of having atleast 1 girlfriend is $\frac{1}{2}$. So $\frac{1}{2}$ mod $10^9 + 7$ is 500000004.

In the 3rd case, total possibilities of distribution are [1,2,3] , [1,3,2], [2,1,3] , [2,3,1] , [3,1,2] ,[3,2,1] . out of the 6 possibilities I will have at least 1 girlfriend in first, second, third, sixth cases . So probability of having atleast 1 girlfriend is $\frac{4}{6}$. So $\frac{4}{6}$ mod $10^9 + 7$ is 666666672.

D. Danger Dungeons

2 seconds, 256 megabytes

Sung-Jin-Woo and Cha-Hae , the powerful S-rank Hunters, were stuck inside a red gate and there was a dungeon inside the gate ...

The dungeon has N monsters and each of them have health of h_i units . The monsters are also S - ranked , so every attack of the hunters reduces the health of a monster only by 1 unit .

Because of the dungeon curse , the following happens :

- Both of them cannot simultaneously attack a monster .
- A monster gets immune to a hunter's attack if he/she takes half of its life . That is to kill a monster, Sung-Jin-Woo should attack half and Cha-Hae should attack half .
- Ex : For a monster of health 6 , Jin-Woo fights till the monster health reduces to 3 and Cha-hae fights till the monster health reduces to 0 and dies... Or first Cha-Hae can fight and then Jin-Woo can take over..
- Monsters cannot regenerate , that is even if Cha - Hae fights a monster and leaves it but Jin-Woo is busy fighting another monster this current monster cannot regain health and Cha -Hae can leave this and move to next monster and Jin-Woo later can come and finish it off....

Assuming the hunters do 1 attack per second , what is the minimum time required to defeat all the monsters and close the Dungeon..

Input

The first line contains an integer N $(1 \le N \le 2.10^5)$.

The next line contains N integers $h_1, h_2, \ldots, h_N$ $(2 \le h_i \le 2.10^9)$ - the health of monsters . It is guaranteed that all h_i 's are even numbers .

Output

Output the minimum time required to defeat all the monsters

input

```
3
2 4 6
```

output

```
6
```

input

```
1
10
```

output

```
10
```

In the first case:

- Initially Jin-Woo fighting with monster 3 and Cha-Hae fighting with monster 2. After t=1s, h = [2,3,5]
- After t=2s, h = [2,2,4]
- Now Cha-Hae reduced the life of 2nd monster by half (initially 4 now 2) so that monster becomes immune to her attack so she leaves that monster and goes to monster 1. After t=3s, h = [1,2,3]
- Now Jin-woo reduced life of 3rd monster by half and Cha-hae reduced the life of 1st monster by half. So Now Jin-woo goes to 1st monster and Cha-hae goes to 3rd monster . After t=4s , h=[0,2,2]
- Jin-woo comes to monster 2, After t=5s, h = [0,1,1]
- After t=6s, h=[0,0,0]
- So after 6 sec all monsters die and Dungeon Closes

In the second test case:

- Let Jin-Woo starts fighting with the monster but Cha-Hae will be waiting till he completes because both cant simultaneously attack the same

monster.
- So after t=5s, monster life will become half (initially 10 now 5).
- So,now it will become immune to Jin-Woo attacks and now Cha-Hae takes over and fights for the next 5 seconds
- After t = 10s, the monster dies and the dungeon is closed..

E. Easy Problem

1 s., 256 MB

Alice and Bob are playing a game. The rules are simple :

- Alice gives bob a number N.
- Now, Bob takes an empty array of size N and fills each position by selecting a random numbers from 1 to k and gives it to Alice. (Any number can occur any number of times in the array but each element will be in the range [1,k] .
- Now in this array , among the all possible subsets, If there exist a subset where all elements in that subset are equal and size of the subset is greater than or equal to x , then Alice Wins , else Bob wins ...

Both of them always want to win... So for a given x and k, what is the minimum N Alice should give Bob so that she can guarantee a win...

Note : An array B is a subset of array A only if B can be obtained from A by deleting several(possibly zero) numbers from A and combining the remaining elements . Ex: [1,3] is a subset of [1,2,3,3] but [1,2,2] is not .

Input

The first line contains a integer t $(1 \leq t \leq 10^5)$ - the number of test cases.

Each test contains 2 integers x and k $(1 \leq x, k \leq 10^9)$ - the min size of subset required and the numbers used for filling the array by Bob.

Output

For each x, k output the minimum N that guarantees Alice's Win .

input
```
2
1 5
3 3
```
output
```
1
7
```

For the first test case, x=1 so Alice want to select only a subset of size 1 , so Alice can ask only for 1 number and select it to achieve the conditions

For the second case, if Alice asks array of size less than 7 , then Bob can always make a array that does not satisfy those conditions and Alice loses.. Ex: If Alice asks N=5 Bob gives her [1,3,2,3,1] , so Alice cannot select a subset of size 3 containing same numbers and Alice Loses.

F. Furious Fermat

2 seconds, 256 megabytes

Ohh no... Our little Fermat is bored and again comes up with a deadly theorem... Mind Blowing...

He is not happy because people easily find the value of $a^b \ mod \ 10^9 + 7$, where $(1 \leq a, b \leq 10^9)$. So now he comes up with this deadly problem.

Find the value of $a^{b^c} \ mod \ 10^9 + 7$, where $(1 \leq a, b, c \leq 10^9)$

Input

First line contains an integer t $(1 \leq t \leq 10^5)$ - the number of test cases .

Each test case contains 3 integers a, b, c where $(1 \leq a, b, c \leq 10^9)$.

Output

Print the value of $a^{b^c} \ mod \ 10^9 + 7$

input
```
3
3 2 3
100 1000 10000
1000000000 1000000000 1000000000
```
output
```
6561
388998981
497489713
```

```
497485715
```

G. Greedy Goat

2 s., 256 MB

The Goat of CP Land is very greedy that it can eat any number of leaves it want... But the goat has its own rules of eating... A typical foodie right ???

There are N pots and each pot has A_i leaves in it . For each second, among the N pots, the goat chooses the pot with the maximum number of leaves and eats half of the leaves from it . That is if A_i is even then it eats $\frac{A_i}{2}$ leaves from that pot in 1 sec or if A_i is odd then it eats $\frac{A_i+1}{2}$ leaves from that pot in 1 sec.. If the maximum number of leaves is present in two or more pots then the goat chooses the leftmost pot from those pots and eats half from it..

I am the owner of the goat and I want to know after how many seconds each pot gets empty so that I can refill it.. But there are too many pots and a hell lots of leaves in it and it is difficult for me to calculate... So please help me find after how many seconds each pot gets empty..

Input

The first line of input contains a number t $(1 \leq t \leq 5.10^4)$ - the number of test cases.

Then the first line of each test case has a number N ($1 \leq N \leq 2.10^5$) - the number of pots .

The next lines contains N integers $A_1, A_2, \ldots\ldots, A_N$ $(1 \leq A_i \leq 10^{18})$ - the number of leaves in each pot .

It is guaranteed that the sum of N over all test cases does not exceed 5.10^5.

Output

For each test case output N integers - where the i-th number denotes the time after which that pot becomes empty .

input

```
3
1
13
2
1 1
2
10000000000 1000000000000
```

output

```
4
1 2
73 74
```

For the first case: There is only 1 pot , so the goat continues eating till it becomes empty.

- The pot initially has 13 leaves. So the goat eats $\frac{13+1}{2} = 7$ leaves and after t = 1sec , 6 leaves remain in the pot .
- Now the pot has 6 leaves. So the goat eats $\frac{6}{2} = 3$ leaves and after t = 2sec , 3 leaves remain in the pot .
- Now the pot has 3 leaves. So the goat eats $\frac{3+1}{2} = 2$ leaves and after t = 3sec , 1 leaf remain in the pot .
- Now the pot has 1 leaf. So the goat eats $\frac{1+1}{2} = 1$ leaf and after t = 4sec , 0 leaves remain in the pot .

So it takes 4 seconds for the pot to become empty.
For the second case:

The maximum number of leaves among the pots is 1 and it occurs in 2 pots , so the goat chooses the leftmost pot and eats half from it first and then it goes for the next pot .

H. Harry Hates DP

1 s., 256 MB

Harry Potter is hiking in the Alps surrounding Lake Geneva. In this area there are m cabins, numbered 1 to m. Each cabin is connected, with one or more trails, to a central meeting point next to the lake. Each trail is either short or long. Cabin i is connected with s_i short trails and l_i long trails to the lake.

Each day, Harry walks a trail from the cabin where he currently is to Lake Geneva, and then from there he walks a trail to any of the m cabins (including the one he started in). However, as he has to finish the hike in a day, at least one of the two trails has to be short.

How many possible combinations of trails can Harry take if he starts in cabin 1 and walks for n days?

Give the answer modulo $10^9 + 7$.

Input

The first line contains the integers m and n.

The second line contains m integers, $s_1, \ldots, s_m$, where s_i is the number of short trails between cabin i and Lake Geneva.

The third and last line contains m integers, $l_1, \ldots, l_m$, where l_i is the number of long trails between cabin i and Lake Geneva.

We have the following constraints:

$0 \leq s_i, l_i \leq 10^3$.

$1 \leq m \leq 10^2$.

$1 \leq n \leq 10^3$.

Output

The number of possible combinations of trails, modulo $10^9 + 7$.

input

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

output

```
18
```