
OPI
Salchichas
Viena Largas
6
unidades
Peso neto:
276g
OPI
Salchichas
tipo Frankfurt
10
unidades
Peso neto:
276g


OPI
Medallones
de Carne Vacuna
4
unidades
Peso neto:
276g

A. Covid Country

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Covid cases are surging in the Country of NLogônia, the Prime Minister has fired all health and statistics officials and now he needs your help. Each day you'll receive two types of queries:

$1, id, x$

You receive the city's id and how many new cases it has.

$2, k$

You have to answer how many cities have more than k cases. **Initially all cities have zero cases.**

Input

The first line contains two integers $(1 \leq q \leq 10^5; 1 \leq n \leq 10^5)$, the number of queries and the number of cities.

q queries follow, each query is in the formats:

`1 id x` where $(0 \leq id < n; 1 \leq x \leq 10^{13})$

`2 k` where $(0 \leq k \leq 10^{18})$

Output

For the second type of query print how many cities have **more than** k cases.

Examples

input

```
6 10
1 0 10
1 1 9
2 2
2 8
2 9
2 10
```

output

```
2
2
1
0
```

input

```
4 5
1 0 10
2 10
1 0 1
2 10
```

output

```
0
1
```

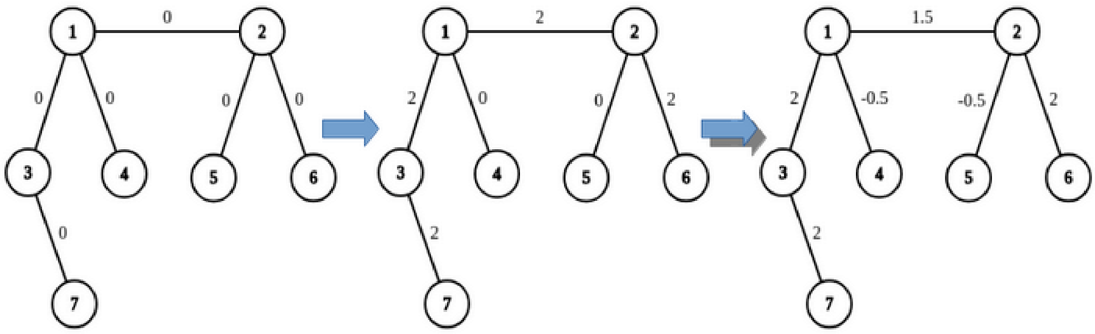
B. Add on a Tree

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Note that this is the first problem of the two similar problems. You can hack this problem only if you solve both problems.

You are given a tree with n nodes. In the beginning, 0 is written on all edges. In one operation, you can choose any 2 distinct **leaves** u, v and any **real** number x and add x to values written on all edges on the simple path between u and v.

For example, on the picture below you can see the result of applying two operations to the graph: adding 2 on the path from 7 to 6, and then adding -0.5 on the path from 4 to 5.



Is it true that for any configuration of real numbers written on edges, we can achieve it with a finite number of operations?

Leaf is a node of a tree of degree 1. Simple path is a path that doesn't contain any node twice.

Input

The first line contains a single integer n ($2 \le n \le 10^5$) — the number of nodes.

Each of the next $n-1$ lines contains two integers u and v ($1 \le u, v \le n$, $u \neq v$), meaning that there is an edge between nodes u and v. It is guaranteed that these edges form a tree.

Output

If there is a configuration of real numbers written on edges of the tree that we can't achieve by performing the operations, output "NO".

Otherwise, output "YES".

You can print each letter in any case (upper or lower).

Examples

input

```
2
1 2
```

output

```
YES
```

input

```
3
1 2
2 3
```

output

```
NO
```

input

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

output

```
NO
```

input

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

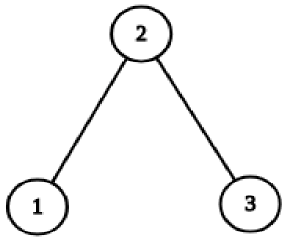
output

```
YES
```

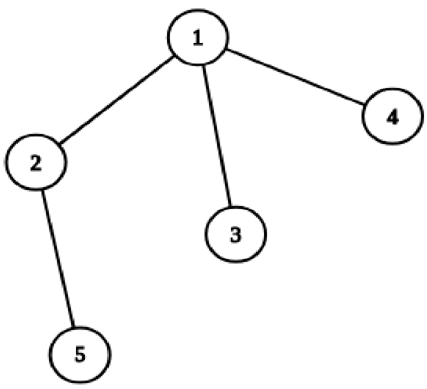
Note

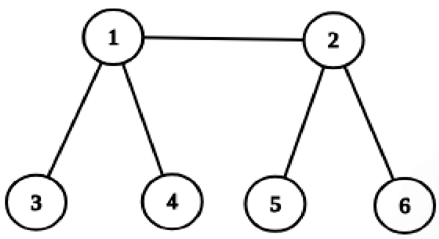
In the first example, we can add any real x to the value written on the only edge $(1, 2)$.

In the second example, one of configurations that we can't reach is 0 written on $(1, 2)$ and 1 written on $(2, 3)$.



Below you can see graphs from examples 3, 4:





C. Zuhair and Strings

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Given a string s of length n and integer k ($1 \le k \le n$). The string s has a level x, if x is largest non-negative integer, such that it's possible to find in s:

- x **non-intersecting** (non-overlapping) substrings of length k,
- all characters of these x substrings are the same (i.e. each substring contains only one distinct character and this character is the same for all the substrings).

A substring is a sequence of consecutive (adjacent) characters, it is defined by two integers i and j ($1 \le i \le j \le n$), denoted as $s[i \dots j]$ = "$s_i s_{i+1} \dots s_j$".

For example, if $k = 2$, then:

- the string "`aabb`" has level 1 (you can select substring "`aa`"),
- the strings "`zzzz`" and "`zzbzz`" has level 2 (you can select two non-intersecting substrings "`zz`" in each of them),
- the strings "`abed`" and "`aca`" have level 0 (you can't find at least one substring of the length $k = 2$ containing the only distinct character).

Zuhair gave you the integer k and the string s of length n. You need to find x, the level of the string s.

Input

The first line contains two integers n and k ($1 \le k \le n \le 2 \cdot 10^5$) — the length of the string and the value of k.

The second line contains the string s of length n consisting only of lowercase Latin letters.

Output

Print a single integer x — the level of the string.

Examples

input Copy

```
8 2
aaacaabb
```

output Copy

```
2
```

input Copy

```
2 1
ab
```

output Copy

```
1
```

input Copy

```
4 2
abab
```

output Copy

```
0
```

Note

In the first example, we can select 2 non-intersecting substrings consisting of letter '`a`': "`(aa)ac(aa)bb`", so the level is 2.

In the second example, we can select either substring "`a`" or "`b`" to get the answer 1.

D. Generate Login

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

The preferred way to generate user login in Polygon is to concatenate a prefix of the user's first name and a prefix of their last name, in that order. Each prefix must be non-empty, and any of the prefixes can be the full name. Typically there are multiple possible logins for each person.

You are given the first and the last name of a user. Return the alphabetically earliest login they can get (regardless of other potential Polygon users).

As a reminder, a prefix of a string s is its substring which occurs at the beginning of s: "a", "ab", "abc" etc. are prefixes of string "{abcdef}" but "b" and 'bc" are not. A string a is alphabetically earlier than a string b, if a is a prefix of b, or a and b coincide up to some position, and then a has a letter that is alphabetically earlier than the corresponding letter in b: "a" and "ab" are alphabetically earlier than "ac" but "b" and "ba" are alphabetically later than "ac".

Input

The input consists of a single line containing two space-separated strings: the first and the last names. Each character of each string is a lowercase English letter. The length of each string is between 1 and 10, inclusive.

Output

Output a single string — alphabetically earliest possible login formed from these names. The output should be given in lowercase as well.

Examples

input Copy

```
harry potter
```

output Copy

```
hap
```

input Copy

```
tom riddle
```

output Copy

```
tomr
```

E. A Tale of Two Lands

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

The legend of the foundation of Vectorland talks of two integers x and y. Centuries ago, the array king placed two markers at points $|x|$ and $|y|$ on the number line and conquered all the land in between (including the endpoints), which he declared to be Arrayland. Many years later, the vector king placed markers at points $|x-y|$ and $|x+y|$ and conquered all the land in between (including the endpoints), which he declared to be Vectorland. He did so in such a way that the land of Arrayland was completely inside (including the endpoints) the land of Vectorland.

Here $|z|$ denotes the absolute value of z.

Now, Jose is stuck on a question of his history exam: "What are the values of x and y?" Jose doesn't know the answer, but he believes he has narrowed the possible answers down to n integers $a_1, a_2, \dots, a_n$. Now, he wants to know the number of **unordered** pairs formed by two **different** elements from these n integers such that the legend could be true if x and y were equal to these two values. Note that it is possible that Jose is wrong, and that no pairs could possibly make the legend true.

Input

The first line contains a single integer n ($2 \le n \le 2 \cdot 10^5$) — the number of choices.

The second line contains n pairwise distinct integers $a_1, a_2, \dots, a_n$ ($-10^9 \le a_i \le 10^9$) — the choices Jose is considering.

Output

Print a single integer number — the number of unordered pairs $\{x, y\}$ formed by different numbers from Jose's choices that could make the legend true.

Examples

input Copy

```
3
2 5 -3
```

output Copy

```
2
```

input Copy

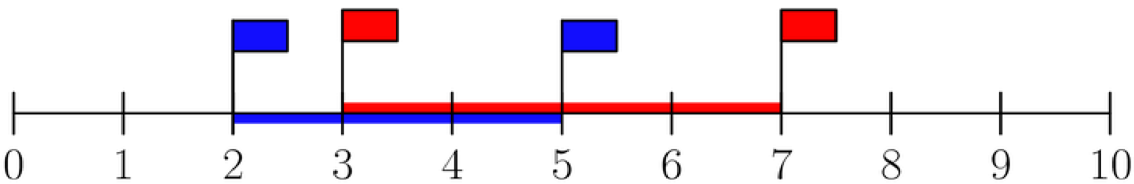
```
2
3 6
```

output Copy

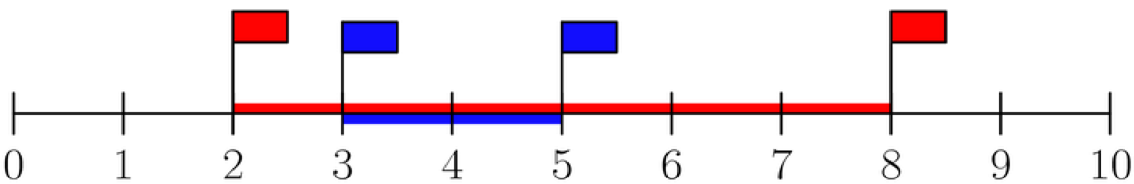
```
1
```

Note

Consider the first sample. For the pair $\{2, 5\}$, the situation looks as follows, with the Arrayland markers at $|2| = 2$ and $|5| = 5$, while the Vectorland markers are located at $|2-5| = 3$ and $|2+5| = 7$:

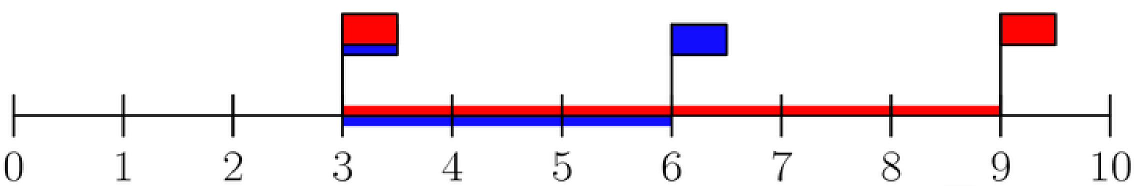


The legend is not true in this case, because the interval $[2, 3]$ is not conquered by Vectorland. For the pair $\{5, -3\}$ the situation looks as follows, with Arrayland consisting of the interval $[3, 5]$ and Vectorland consisting of the interval $[2, 8]$:



As Vectorland completely contains Arrayland, the legend is true. It can also be shown that the legend is true for the pair $\{2, -3\}$, for a total of two pairs.

In the second sample, the only pair is $\{3, 6\}$, and the situation looks as follows:



Note that even though Arrayland and Vectorland share 3 as endpoint, we still consider Arrayland to be completely inside of Vectorland.

F. Vanya and Lanterns

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output

Vanya walks late at night along a straight street of length l, lit by n lanterns. Consider the coordinate system with the beginning of the street corresponding to the point 0, and its end corresponding to the point l. Then the i-th lantern is at the point a_i. The lantern lights all points of the street that are at the distance of at most d from it, where d is some positive number, common for all lanterns.

Vanya wonders: what is the minimum light radius d should the lanterns have to light the whole street?

Input

The first line contains two integers n, l ($1 \le n \le 1000$, $1 \le l \le 10^9$) — the number of lanterns and the length of the street respectively.

The next line contains n integers a_i ($0 \le a_i \le l$). Multiple lanterns can be located at the same point. The lanterns may be located at the ends of the street.

Output

Print the minimum light radius d, needed to light the whole street. The answer will be considered correct if its absolute or relative error doesn't exceed 10^{-9}.

Examples

input

```
7 15
15 5 3 7 9 14 0
```

output

```
2.5000000000
```

input

```
2 5
2 5
```

output

```
2.0000000000
```

Note

Consider the second sample. At $d = 2$ the first lantern will light the segment $[0, 4]$ of the street, and the second lantern will light segment $[3, 5]$. Thus, the whole street will be lit.

G. Police Hypothesis

time limit per test: 8 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

The public transport system of Nlogônia has an express network connecting the main points of interest of the country. There are $N-1$ bullet trains connecting N attractions such that from one of the points of interest you can reach any other using only this network.

As anywhere in the world, it is common that there is graffiti on the train stations. What caught the attention of the police of the country is the fact that in each one of the stations it is possible to find exactly one letter pinned with a specific style. The hypothesis is that criminals may be changing the graffiti as a means of communication and, therefore, it was decided to create a system capable of monitoring the graffiti and its amendments.

Given a pattern P, the description of the connections between stations and the suspicious letters in each of them, your task is to write a program able to deal with the following operations:

- `1` u v: print how many times the pattern P occurs in the path from u to v if we look at the string of characters associated with the consecutive vertices of the path;
- `2` u x: change the suspicious letter at the station u to x.

Input

The first input line contains two integers N and Q $(1 \le N, Q \le 10^5)$, representing the number of stations and the number of transactions that must be processed. The second line contains the pattern P monitored $(1 \le |P| \le 100)$. The third line contains a string with N characters representing the letters initially associated with each of the stations. Each of the following $N-1$ lines contains two integers u and v indicating that there is a bullet train between u and v. The following Q lines describe the operations that must be processed as described above.

Output

Your program should print a line for each operation of type 1 containing an integer that represents the number of occurrences of the pattern P on the way.

Examples

input

```
4 4
xtc
xtzy
1 2
2 3
3 4
1 1 3
2 3 c
1 1 3
1 3 1
```

output

```
0
1
0
```

input

```
6 7
lol
dlorlx
1 2
1 3
3 4
3 5
5 6
1 2 6
2 3 l
2 6 l
2 5 o
1 2 6
2 1 o
1 6 2
```

output

```
0
1
2
```

input

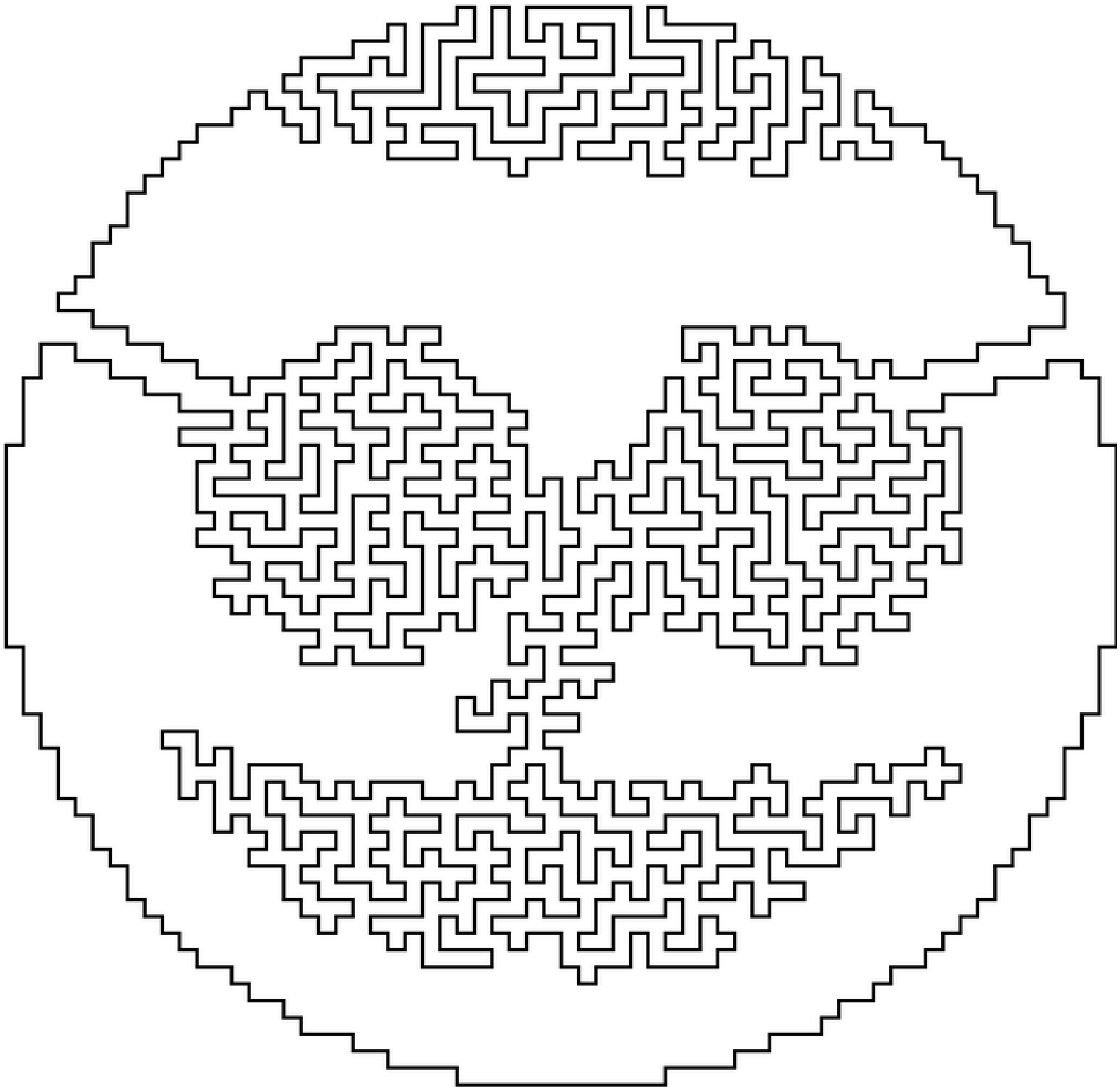
```
5 2
aba
ababa
1 2
2 3
3 4
4 5
1 1 5
1 5 1
```

output

```
2
2
```

H. Jordan Smiley

time limit per test: 1 second
memory limit per test: 256 megabytes
input: standard input
output: standard output



Input

The input contains two integers row, col ($0 \le row, col \le 63$), separated by a single space.

Output

Output "IN" or "OUT".

Examples

input

```
0 0
```

output

```
OUT
```

input

```
27 0
```

output

```
IN
```

input

```
0 27
```

output

```
OUT
```

input

```
27 27
```

output

```
IN
```

I. Gasoline

time limit per test: 0.5 seconds
memory limit per test: 1024 megabytes
input: standard input
output: standard output

After the end of the truck drivers' strike, you and the rest of Nlogônia logistics specialists now have the task of planning the refueling of the gas stations in the city. For this, we collected information on stocks of R refineries and about the demands of P gas stations. In addition, there are contractual restrictions that some refineries cannot supply some gas stations; When a refinery can provide a station, the shorter route to transport fuel from one place to another is known.

The experts' task is to minimize the time all stations are supplied, satisfying their demands. The refineries have a sufficiently large amount of trucks, so that you can assume that each truck will need to make only one trip from a refinery to a gas station. The capacity of each truck is greater than the demand of any gas station, but it may be necessary to use more than one refinery.

Input

The first line of the input contains three integers P, R, C, respectively the number of gas stations, the number of refineries and the number of pairs of refineries and gas stations whose time will be given ($1 \leq P, R \leq 1000$; $1 \leq C \leq 20000$). The second line contains P integers D_i ($1 \leq D_i \leq 10^4$), representing the demands in liters of gasoline of the gas stations $i = 1, 2, \ldots, P$, in that order. The third line contains R integers E_i ($1 \leq E_i \leq 10^4$), representing stocks, in liters of gasoline, of refineries $i = 1, 2, \ldots, R$, in that order. Finally, the latest C lines describe course times, in minutes, between stations and refineries. Each of these rows contains three integers, I, J, T ($1 \leq I \leq P$; $1 \leq J \leq R$; $1 \leq T \leq 10^6$), where I is the ID of a post, J is the ID of a refinery and T is the time in the course of a refinery truck J to I. No pair (J, I) repeats. Not all pairs are informed; If a pair is not informed, contractual restrictions prevents the refinery from supplying the station.

Output

Print an integer that indicates the minimum time in minutes for all stations to be completely filled up. If this is not possible, print -1.

Examples

input

```
3 2 5
20 10 10
30 20
1 1 2
2 1 1
2 2 3
3 1 4
3 2 5
```

output

```
4
```

input

```
3 2 5
20 10 10
25 30
1 1 3
2 1 1
2 2 4
3 1 2
3 2 5
```

output

```
5
```

input

```
4 3 9
10 10 10 20
10 15 30
1 1 1
1 2 1
2 1 3
2 2 2
3 1 10
3 2 10
4 1 1
4 2 2
4 3 30
```

output

```
-1
```

input

```
1 2 2
40
30 10
1 1 100
1 2 200
```

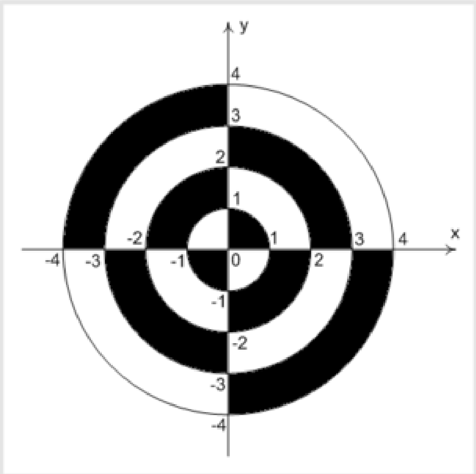
output

```
200
```

J. Find Color

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Not so long ago as a result of combat operations the main Berland place of interest — the magic clock — was damaged. The cannon's balls made several holes in the clock, that's why the residents are concerned about the repair. The magic clock can be represented as an **infinite** Cartesian plane, where the origin corresponds to the clock center. The clock was painted two colors as is shown in the picture:



The picture shows only the central part of the clock. This coloring naturally extends to infinity.

The balls can be taken to be points on the plane. Your task is to find the color of the area, damaged by the given ball.

All the points located on the border of one of the areas have to be considered painted black.

Input

The first and single line contains two integers x and y — the coordinates of the hole made in the clock by the ball. Each of the numbers x and y has an absolute value that does not exceed 1000.

Output

Find the required color.

All the points between which and the origin of coordinates the distance is integral-value are painted black.

Examples

input
-2 1

output
white

input
2 1

output
black

input
4 3

output
black

K. Arena of Greed

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

Lately, Mr. Chanek frequently plays the game **Arena of Greed**. As the name implies, the game's goal is to find the greediest of them all, who will then be crowned king of Compfestnesia.

The game is played by two people taking turns, where Mr. Chanek takes the first turn. Initially, there is a treasure chest containing N gold coins. The game ends if there are no more gold coins in the chest. In each turn, the players can make one of the following moves:

- Take one gold coin from the chest.
- Take half of the gold coins on the chest. This move is only available if the number of coins in the chest is even.

Both players will try to maximize the number of coins they have. Mr. Chanek asks your help to find the maximum number of coins he can get at the end of the game if both he and the opponent plays optimally.

Input

The first line contains a single integer T $(1 \le T \le 10^5)$ denotes the number of test cases.

The next T lines each contain a single integer N $(1 \le N \le 10^{18})$.

Output

T lines, each line is the answer requested by Mr. Chanek.

Example

input

```
2
5
6
```

output

```
2
4
```

Note

For the first case, the game is as follows:

1. Mr. Chanek takes one coin.
2. The opponent takes two coins.
3. Mr. Chanek takes one coin.
4. The opponent takes one coin.

For the second case, the game is as follows:

1. Mr. Chanek takes three coins.
2. The opponent takes one coin.
3. Mr. Chanek takes one coin.
4. The opponent takes one coin.