Codeforces Round #457 (Div. 2)

A. Jamie and Alarm Snooze

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

Jamie loves sleeping. One day, he decides that he needs to wake up at exactly $hh:mm$. However, he hates waking up, so he wants to make waking up less painful by setting the alarm at a `lucky` time. He will then press the snooze button every x minutes until $hh:mm$ is reached, and only then he will wake up. He wants to know what is the smallest number of times he needs to press the snooze button.

A time is considered `lucky` if it contains a digit '7'. For example, $13:07$ and $17:27$ are `lucky`, while $00:48$ and $21:34$ are not `lucky`.

Note that it is not necessary that the time set for the alarm and the wake-up time are on the same day. It is guaranteed that there is a `lucky` time Jamie can set so that he can wake at $hh:mm$.

Formally, find the smallest possible non-negative integer y such that the time representation of the time $x \cdot y$ minutes before $hh:mm$ contains the digit '7'.

Jamie uses 24-hours clock, so after $23:59$ comes $00:00$.

Input

The first line contains a single integer x ($1 \le x \le 60$).

The second line contains two two-digit integers, hh and mm ($00 \le hh \le 23$, $00 \le mm \le 59$).

Output

Print the minimum number of times he needs to press the button.

Examples

input

```
3
11 23
```

output

```
2
```

input

```
5
01 07
```

output

```
0
```

Note

In the first sample, Jamie needs to wake up at 11:23. So, he can set his alarm at 11:17. He would press the snooze button when the alarm rings at 11:17 and at 11:20.

In the second sample, Jamie can set his alarm at exactly at 01:07 which is `lucky`.

B. Jamie and Binary Sequence (changed after round)

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

Jamie is preparing a Codeforces round. He has got an idea for a problem, but does not know how to solve it. Help him write a solution to the following problem:

Find k integers such that the sum of two to the power of each number equals to the number n and the largest integer in the answer is as small as possible. As there may be multiple answers, you are asked to output the lexicographically largest one.

To be more clear, consider all integer sequence with length k $(a_1, a_2, ..., a_k)$ with $\sum_{i=1}^{k} 2^{a_i} = n$. Give a value $y = \max_{1 \le i \le k} a_i$ to each sequence. Among all sequence(s) that have the minimum y value, output the one that is the lexicographically largest.

For definitions of powers and lexicographical order see notes.

Input

The first line consists of two integers n and k $(1 \le n \le 10^{18}, 1 \le k \le 10^5)$ — the required sum and the length of the sequence.

Output

Output "No" (without quotes) in a single line if there does not exist such sequence. Otherwise, output "Yes" (without quotes) in the first line, and k numbers separated by space in the second line — the required sequence.

It is guaranteed that the integers in the answer sequence fit the range $[-10^{18}, 10^{18}]$.

Examples

input Copy

```
23 5
```

output

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

input Copy

```
13 2
```

output

```
No
```

input Copy

```
1 2
```

output

```
Yes
-1 -1
```

Note

Sample 1:

$2^3 + 2^3 + 2^2 + 2^1 + 2^0 = 8 + 8 + 4 + 2 + 1 = 23$

Answers like $(3, 3, 2, 0, 1)$ or $(0, 1, 2, 3, 3)$ are not lexicographically largest.

Answers like $(4, 1, 1, 1, 0)$ do not have the minimum y value.

Sample 2:

It can be shown there does not exist a sequence with length 2.

Sample 3:

$2^{-1} + 2^{-1} = \frac{1}{2} + \frac{1}{2} = 1$

Powers of 2:

If $x > 0$, then $2^x = 2 \cdot 2 \cdot 2 \cdot ... \cdot 2$ (x times).

If $x = 0$, then $2^x = 1$.

If $x < 0$, then $2^x = \frac{1}{2^{-x}}$.

Lexicographical order:

Given two different sequences of the same length, $(a_1, a_2, \ldots , a_k)$ and $(b_1, b_2, \ldots , b_k)$, the first one is smaller than the second one for the lexicographical order, if and only if $a_i < b_i$, for the first i where a_i and b_i differ.

C. Jamie and Interesting Graph

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

Jamie has recently found undirected weighted graphs with the following properties very `interesting`:

- The graph is connected and contains exactly n vertices and m edges.
- All edge weights are integers and are in range $[1, 10^9]$ inclusive.
- The length of shortest path from 1 to n is a prime number.
- The sum of edges' weights in the minimum spanning tree (MST) of the graph is a prime number.
- The graph contains no loops or multi-edges.

If you are not familiar with some terms from the statement you can find definitions of them in notes section.

Help Jamie construct any graph with given number of vertices and edges that is `interesting`!

Input

First line of input contains 2 integers n, m $(2 \leq n \leq 10^5, n-1 \leq m \leq min(\frac{n(n-1)}{2}, 10^5))$ — the required number of vertices and edges.

Output

In the first line output 2 integers $sp, mstw$ $(1 \leq sp, mstw \leq 10^{14})$ — the length of the shortest path and the sum of edges' weights in the minimum spanning tree.

In the next m lines output the edges of the graph. In each line output 3 integers u, v, w $(1 \leq u, v \leq n, 1 \leq w \leq 10^9)$ describing the edge connecting u and v and having weight w.

Examples

input

```
4 4
```

output

```
7 7
1 2 3
2 3 2
3 4 2
2 4 4
```

input

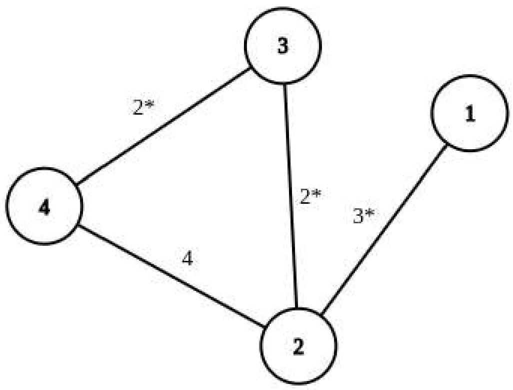
```
5 4
```

output

```
7 13
1 2 2
1 3 4
1 4 3
4 5 4
```

Note

The graph of sample 1:



Shortest path sequence: $\{1, 2, 3, 4\}$. MST edges are marked with an asterisk (*).

Definition of terms used in the problem statement:

A **shortest path** in an undirected graph is a sequence of vertices $(v_1, v_2, \ldots, v_k)$ such that v_i is adjacent to v_{i+1} $1 \le i < k$ and the sum of weight $\sum_{i=1}^{n-1} w(v_i, v_{i+1})$ is minimized where $w(i, j)$ is the edge weight between i and j. (https://en.wikipedia.org/wiki/Shortest_path_problem)

A **prime number** is a natural number greater than 1 that has no positive divisors other than 1 and itself. (https://en.wikipedia.org/wiki/Prime_number)

A **minimum spanning tree (MST)** is a subset of the edges of a connected, edge-weighted undirected graph that connects all the vertices together, without any cycles and with the minimum possible total edge weight. (https://en.wikipedia.org/wiki/Minimum_spanning_tree)

https://en.wikipedia.org/wiki/Multiple_edges

D. Jamie and To-do List

time limit per test: 2 seconds
memory limit per test: 512 megabytes
input: standard input
output: standard output

Why I have to finish so many assignments???

Jamie is getting very busy with his school life. He starts to forget the assignments that he has to do. He decided to write the things down on a to-do list. He assigns a value `priority` for each of his assignment **(lower value means more important)** so he can decide which he needs to spend more time on.

After a few days, Jamie finds out the list is too large that he can't even manage the list by himself! As you are a good friend of Jamie, help him write a program to support the following operations on the to-do list:

- $set\ a_i\ x_i$ — Add assignment a_i to the to-do list if it is not present, and set its `priority` to x_i. If assignment a_i is already in the to-do list, its `priority` **is changed** to x_i.
- $remove\ a_i$ — Remove assignment a_i from the to-do list if it is present in it.
- $query\ a_i$ — Output the number of assignments that are more important (have a **smaller** `priority` value) than assignment a_i, so Jamie can decide a better schedule. Output -1 if a_i is not in the to-do list.
- $undo\ d_i$ — Undo all changes that have been made in the previous d_i days (not including the day of this operation)

At day 0, the to-do list is empty. In each of the following q days, Jamie will do **exactly one** out of the four operations. If the operation is a $query$, you should **output the result of the query before proceeding to the next day**, or poor Jamie cannot make appropriate decisions.

Input

The first line consists of a single integer q $(1 \le q \le 10^5)$ — the number of operations.

The following q lines consists of the description of the operations. The i-th line consists of the operation that Jamie has done in the i-th day. The query has the following format:

The first word in the line indicates the type of operation. It must be one of the following four: `set`, `remove`, `query`, `undo`.

- If it is a `set` operation, a string a_i and an integer x_i follows $(1 \le x_i \le 10^9)$. a_i is the assignment that need to be set to `priority` x_i.
- If it is a `remove` operation, a string a_i follows. a_i is the assignment that need to be removed.
- If it is a `query` operation, a string a_i follows. a_i is the assignment that needs to be queried.
- If it is a `undo` operation, an integer d_i follows $(0 \le d_i < i)$. d_i is the number of days that changes needed to be undone.

All assignment names a_i only consists of lowercase English letters and have a length $1 \le |a_i| \le 15$.

It is guaranteed that the last operation is a `query` operation.

Output

For each `query` operation, output a single integer — the number of assignments that have a priority lower than assignment a_i, or -1 if a_i is not in the to-do list.

Interaction

If the operation is a $query$, you **should** output the result of the query and flush the output stream before proceeding to the next operation. Otherwise, you may get the verdict `Idleness Limit Exceed`.

For flushing the output stream, please refer to the documentation of your chosen programming language. The flush functions of some common programming languages are listed below:

- C: `fflush(stdout);`
- C++: `cout « flush;`
- Java: `System.out.flush();`

Examples

input

```
8
set chemlabreport 1
set physicsexercise 2
set chinesemockexam 3
query physicsexercise
query chinesemockexam
remove physicsexercise
query physicsexercise
query chinesemockexam
```

output

```
1
2
-1
1
```

input

```
8
set physicsexercise 2
set chinesemockexam 3
set physicsexercise 1
query physicsexercise
query chinesemockexam
undo 4
query physicsexercise
query chinesemockexam
```

output

```
0
1
0
-1
```

input

```
5
query economicsessay
remove economicsessay
query economicsessay
undo 2
query economicsessay
```

output

```
-1
-1
-1
```

input

```
5
set economicsessay 1
remove economicsessay
undo 1
undo 1
query economicsessay
```

output

```
-1
```

E. Jamie and Tree

time limit per test: 2.5 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

To your surprise, Jamie is the final boss! Ehehehe.

Jamie has given you a tree with n vertices, numbered from 1 to n. Initially, the root of the tree is the vertex with number 1. Also, each vertex has a value on it.

Jamie also gives you three types of queries on the tree:

$1\ v$ — Change the tree's root to vertex with number v.

$2\ u\ v\ x$ — For each vertex in the subtree of smallest size that contains u and v, add x to its value.

$3\ v$ — Find sum of values of vertices in the subtree of vertex with number v.

A subtree of vertex v is a set of vertices such that v lies on shortest path from this vertex to root of the tree. Pay attention that subtree of a vertex can change after changing the tree's root.

Show your strength in programming to Jamie by performing the queries accurately!

Input

The first line of input contains two space-separated integers n and q $(1 \le n \le 10^5, 1 \le q \le 10^5)$ — the number of vertices in the tree and the number of queries to process respectively.

The second line contains n space-separated integers $a_1, a_2, ..., a_n$ $(- 10^8 \le a_i \le 10^8)$ — initial values of the vertices.

Next $n - 1$ lines contains two space-separated integers u_i, v_i $(1 \le u_i, v_i \le n)$ describing edge between vertices u_i and v_i in the tree.

The following q lines describe the queries.

Each query has one of following formats depending on its type:

$1\ v$ $(1 \le v \le n)$ for queries of the first type.

$2\ u\ v\ x$ $(1 \le u, v \le n, - 10^8 \le x \le 10^8)$ for queries of the second type.

$3\ v$ $(1 \le v \le n)$ for queries of the third type.

All numbers in queries' descriptions are integers.

The queries must be carried out in the given order. It is guaranteed that the tree is valid.

Output

For each query of the third type, output the required answer. It is guaranteed that at least one query of the third type is given by Jamie.

Examples

input

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

output

```
27
19
5
```

input

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

```
2 2 4 3
1 1
2 2 4 -3
3 1
```

output

```
18
21
```

Note

The following picture shows how the tree varies after the queries in the first sample.