Codeforces Round #382 (Div. 2)

A. Ostap and Grasshopper

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

On the way to Rio de Janeiro Ostap kills time playing with a grasshopper he took with him in a special box. Ostap builds a line of length n such that some cells of this line are empty and some contain obstacles. Then, he places his grasshopper to one of the empty cells and a small insect in another empty cell. The grasshopper wants to eat the insect.

Ostap knows that grasshopper is able to jump to any empty cell that is **exactly** k cells away from the current (to the left or to the right). Note that it doesn't matter whether intermediate cells are empty or not as the grasshopper makes a jump over them. For example, if $k = 1$ the grasshopper can jump to a neighboring cell only, and if $k = 2$ the grasshopper can jump over a single cell.

Your goal is to determine whether there is a sequence of jumps such that grasshopper will get from his initial position to the cell with an insect.

Input

The first line of the input contains two integers n and k ($2 \le n \le 100$, $1 \le k \le n - 1$) — the number of cells in the line and the length of one grasshopper's jump.

The second line contains a string of length n consisting of characters '`.`', '`#`', '`G`' and '`T`'. Character '`.`' means that the corresponding cell is empty, character '`#`' means that the corresponding cell contains an obstacle and grasshopper can't jump there. Character '`G`' means that the grasshopper starts at this position and, finally, '`T`' means that the target insect is located at this cell. It's guaranteed that characters '`G`' and '`T`' appear in this line **exactly once**.

Output

If there exists a sequence of jumps (each jump of length k), such that the grasshopper can get from his initial position to the cell with the insect, print "`YES`" (without quotes) in the only line of the input. Otherwise, print "`NO`" (without quotes).

Examples

input
```
5 2
#G#T#
```
output
```
YES
```

input
```
6 1
T....G
```
output
```
YES
```

input
```
7 3
T..#..G
```
output
```
NO
```

input
```
6 2
..GT..
```
output
```
NO
```

Note

In the first sample, the grasshopper can make one jump to the right in order to get from cell 2 to cell 4.

In the second sample, the grasshopper is only able to jump to neighboring cells but the way to the insect is free — he can get there by jumping left 5 times.

In the third sample, the grasshopper can't make a single jump.

In the fourth sample, the grasshopper can only jump to the cells with odd indices, thus he won't be able to reach the insect.

B. Urbanization

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

Local authorities have heard a lot about combinatorial abilities of Ostap Bender so they decided to ask his help in the question of urbanization. There are n people who plan to move to the cities. The wealth of the i of them is equal to a_i. Authorities plan to build two cities, first for n_1 people and second for n_2 people. Of course, each of n candidates can settle in only one of the cities. Thus, first some subset of candidates of size n_1 settle in the first city and then some subset of size n_2 is chosen among the remaining candidates and the move to the second city. All other candidates receive an official refuse and go back home.

To make the statistic of local region look better in the eyes of their bosses, local authorities decided to pick subsets of candidates in such a way that **the sum of arithmetic mean** of wealth of people in each of the cities is as large as possible. Arithmetic mean of wealth in one city is the sum of wealth a_i among all its residents divided by the number of them (n_1 or n_2 depending on the city). The division should be done in real numbers without any rounding.

Please, help authorities find the optimal way to pick residents for two cities.

Input

The first line of the input contains three integers n, n_1 and n_2 ($1 \le n, n_1, n_2 \le 100\,000$, $n_1 + n_2 \le n$) — the number of candidates who want to move to the cities, the planned number of residents of the first city and the planned number of residents of the second city.

The second line contains n integers $a_1, a_2, ..., a_n$ ($1 \le a_i \le 100\,000$), the i-th of them is equal to the wealth of the i-th candidate.

Output

Print one real value — the maximum possible sum of arithmetic means of wealth of cities' residents. You answer will be considered correct if its absolute or relative error does not exceed 10^{-6}.

Namely: let's assume that your answer is a, and the answer of the jury is b. The checker program will consider your answer correct, if .

Examples

input

```
2 1 1
1 5
```

output

```
6.00000000
```

input

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

output

```
6.50000000
```

Note

In the first sample, one of the optimal solutions is to move candidate 1 to the first city and candidate 2 to the second.

In the second sample, the optimal solution is to pick candidates 3 and 4 for the first city, and candidate 2 for the second one. Thus we obtain $(a_3 + a_4) / 2 + a_2 = (3 + 2) / 2 + 4 = 6.5$

C. Tennis Championship

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

Famous Brazil city Rio de Janeiro holds a tennis tournament and Ostap Bender doesn't want to miss this event. There will be n players participating, and the tournament will follow knockout rules from the very first game. That means, that if someone loses a game he leaves the tournament immediately.

Organizers are still arranging tournament grid (i.e. the order games will happen and who is going to play with whom) but they have already fixed one rule: two players can play against each other only if the number of games one of them has already played **differs by no more than one** from the number of games the other one has already played. Of course, both players had to win all their games in order to continue participating in the tournament.

Tournament hasn't started yet so the audience is a bit bored. Ostap decided to find out what is the maximum number of games the winner of the tournament can take part in (assuming the rule above is used). However, it is unlikely he can deal with this problem without your help.

Input

The only line of the input contains a single integer n $(2 \le n \le 10^{18})$ — the number of players to participate in the tournament.

Output

Print the maximum number of games in which the winner of the tournament can take part.

Examples

input
```
2
```
output
```
1
```

input
```
3
```
output
```
2
```

input
```
4
```
output
```
2
```

input
```
10
```
output
```
4
```

Note

In all samples we consider that player number 1 is the winner.

In the first sample, there would be only one game so the answer is 1.

In the second sample, player 1 can consequently beat players 2 and 3.

In the third sample, player 1 can't play with each other player as after he plays with players 2 and 3 he can't play against player 4, as he has 0 games played, while player 1 already played 2. Thus, the answer is 2 and to achieve we make pairs $(1, 2)$ and $(3, 4)$ and then clash the winners.

D. Taxes

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

Mr. Funt now lives in a country with a very specific tax laws. The total income of mr. Funt during this year is equal to n ($n \geq 2$) burles and the amount of tax he has to pay is calculated as the maximum divisor of n (not equal to n, of course). For example, if $n = 6$ then Funt has to pay 3 burles, while for $n = 25$ he needs to pay 5 and if $n = 2$ he pays only 1 burle.

As mr. Funt is a very opportunistic person he wants to cheat a bit. In particular, he wants to split the initial n in several parts $n_1 + n_2 + ... + n_k = n$ (here k is arbitrary, even $k = 1$ is allowed) and pay the taxes for each part separately. He can't make some part equal to 1 because it will reveal him. So, the condition $n_i \geq 2$ should hold for all i from 1 to k.

Ostap Bender wonders, how many money Funt has to pay (i.e. minimal) if he chooses and optimal way to split n in parts.

Input

The first line of the input contains a single integer n ($2 \leq n \leq 2 \cdot 10^9$) — the total year income of mr. Funt.

Output

Print one integer — minimum possible number of burles that mr. Funt has to pay as a tax.

Examples

input
```
4
```
output
```
2
```

input
```
27
```
output
```
3
```

E. Ostap and Tree

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

Ostap already settled down in Rio de Janiero suburb and started to grow a tree in his garden. Recall that a tree is a connected undirected acyclic graph.

Ostap's tree now has n vertices. He wants to paint some vertices of the tree black such that from any vertex u there is at least one black vertex v at distance no more than k. *Distance* between two vertices of the tree is the minimum possible number of edges of the path between them.

As this number of ways to paint the tree can be large, Ostap wants you to compute it modulo $10^9 + 7$. Two ways to paint the tree are considered different if there exists a vertex that is painted black in one way and is not painted in the other one.

Input

The first line of the input contains two integers n and k $(1 \le n \le 100, 0 \le k \le min(20, n - 1))$ — the number of vertices in Ostap's tree and the maximum allowed distance to the nearest black vertex. **Don't miss** the unusual constraint for k.

Each of the next $n - 1$ lines contain two integers u_i and v_i $(1 \le u_i, v_i \le n)$ — indices of vertices, connected by the i-th edge. It's guaranteed that given graph is a tree.

Output

Print one integer — the remainder of division of the number of ways to paint the tree by $1\ 000\ 000\ 007$ $(10^9 + 7)$.

Examples

input

```
2 0
1 2
```

output

```
1
```

input

```
2 1
1 2
```

output

```
3
```

input

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

output

```
9
```

input

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

output

```
91
```

Note

In the first sample, Ostap has to paint both vertices black.

In the second sample, it is enough to paint only one of two vertices, thus the answer is 3: Ostap can paint only vertex 1, only vertex 2, vertices 1 and 2 both.

In the third sample, the valid ways to paint vertices are: $\{1, 3\}$, $\{1, 4\}$, $\{2, 3\}$, $\{2, 4\}$, $\{1, 2, 3\}$, $\{1, 2, 4\}$, $\{1, 3, 4\}$, $\{2, 3, 4\}$, $\{1, 2, 3, 4\}$.