Codeforces Round #382 (Div. 1)

A. 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.

B. 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

C. 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: $\{1, 3\}, \{1, 4\}, \{2, 3\}, \{2, 4\}, \{1, 2, 3\}, \{1, 2, 4\}, \{1, 3, 4\}, \{2, 3, 4\}, \{1, 2, 3, 4\}$.

D. Permutations

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

Ostap Bender is worried that people started to forget that he is the Great Combinator. Now he wants to show them his skills in combinatorics. Now he studies the permutations of length n. He has a list of m valid pairs, pair a_i and b_i means that he is allowed to place integers b_i at position a_i.

He knows that the number of permutations that use only valid pairs is **odd**. Now, for each pair he wants to find out, will the number of valid permutations be **odd** if he **removes** this pair (and only it) from the list.

Input

The first line contains two integers n and m ($1 \le n \le 2000$, $n \le m \le min(n^2, 500\,000)$) — the number of elements in the permutation. Then follow m lines, each containing some valid pair (a_i, b_i) ($1 \le a_i, b_i \le n$). It's guaranteed that no pair occurs in the input twice and that the total number of valid permutations (i.e. using only allowed pairs position-elements) is odd.

Output

Print m lines, one line for each valid pair. The i-th line should contain "YES" if after Ostap removes the i-th pair (and only it) the remaining number of valid permutations is odd. Otherwise, print «NO».

Examples

input
2 3 1 1 1 2 2 2
output
NO YES NO

input
3 3 1 1 2 2 3 3
output
NO NO NO

input
3 7 3 3 3 1 1 3 1 1 2 2 1 2 2 1
output
YES NO NO NO YES NO NO

E. Chess Championship

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

Ostap is preparing to play chess again and this time he is about to prepare. Thus, he was closely monitoring one recent chess tournament. There were m players participating and each pair of players played exactly one game. The victory gives 2 points, draw $-$ 1 points, lose $-$ 0 points.

Ostap is lazy, so he never tries to remember the outcome of each game. Instead, he computes the total number of points earned by each of the players (the sum of his points in all games which he took part in), sort these value in non-ascending order and then remembers first n integers in this list.

Now the Great Strategist Ostap wonders whether he remembers everything correct. He considers that he is correct if there exists at least one tournament results table such that it will produce the given integers. That means, if we count the sum of points for each player, sort them and take first n elements, the result will coincide with what Ostap remembers. Can you check if such table exists?

Input

The first line of the input contains two integers m and n ($1 \le n \le m \le 3000$) $-$ the number of participants of the tournament and the number of top results Ostap remembers.

The second line contains n integers, provided in non-ascending order $-$ the number of points earned by top participants as Ostap remembers them. It's guaranteed that this integers are non-negative and do not exceed $2 \cdot m$.

Output

If there is no tournament such that Ostap can obtain the given set of integers using the procedure described in the statement, then print "`no`" in the only line of the output. Otherwise, the first line of the output should contain the word "`yes`". Next m lines should provide the description of any valid tournament. Each of these lines must contain m characters '`X`', '`W`', '`D`' and '`L`'. Character '`X`' should always be located on the main diagonal (and only there), that is on the i-th position of the i-th string. Character '`W`' on the j-th position of the i-th string means that the i-th player won the game against the j-th. In the same way character '`L`' means loose and '`D`' means draw.

The table you print must be consistent and the points earned by best n participants should match the memory of Ostap. If there are many possible answers, print any of them.

Examples

input
5 5 8 6 4 2 0
output
yes XWWWW LXWWW LLXWW LLLXW LLLLX

input
5 1 9
output
no