Problem A. Subordinates

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

The company has n employees, numbered $1, 2, \ldots, n$. Employee 1 is the general director of the company. Employees $2, 3, \ldots, n$ reports to a direct manager in the company. Given the structure of a company, your task is to calculate for each employee the number of their subordinates.

Input

The first line has an integer n ($1 \leq n \leq 2 \cdot 10^5$). The second line has $n-1$ integers: the direct manager for each employee $2, 3, \ldots, n$ in the company.

Output

Print n integers: for each employee $1, 2, \ldots, n$ the number of their subordinates.

Examples

Standard Input	Standard Output
5 1 1 2 3	4 1 1 0 0

Problem B. Road Reparation

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

There are n cities and m roads between them. Unfortunately, the condition of the roads is so poor that they cannot be used. Your task is to repair some of the roads so that there will be a decent route between any two cities.

For each road, you know its reparation cost, and you should find a solution where the total cost is as small as possible.

Input

The first input line has two integers n and m: the number of cities and roads. The cities are numbered $1, 2, \ldots, n$.

Then, there are m lines describing the roads. Each line has three integers a, b and c: there is a road between cities a and b, and its reparation cost is c. All roads are two-way roads.

Every road is between two different cities, and there is at most one road between two cities.

- $1 \leq n \leq 10^5$
- $1 \leq m \leq 2 \times 10^5$
- $1 \leq a, b \leq n$
- $1 \leq c \leq 10^9$

Output

Print one integer: the minimum total reparation cost.

However, if there are no solutions, print "IMPOSSIBLE".

Examples

Standard Input	Standard Output
5 6 1 2 3 2 3 5 2 4 2 3 4 8 5 1 7 5 4 4	14
10 10 7 9 3 6 10 4 3 5 7 8 10 6 3 7 3 5 9 2 2 4 4 1 3 5 4 6 9 2 6 2	IMPOSSIBLE

Problem C. Longest Increasing Subsequence

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

You are given an array of n integers. Your task is to determine the length of the longest increasing subsequence in the array.

Note that a subsequence of an array can be obtained by deletion of several (or zero) elements in the array. An increasing subsequence requires each element (except the first element) to be strictly larger than the previous one.

Input

The first line contains an integer n ($1 \leq n \leq 2 \cdot 10^5$): the size of the array.

The second line has n integers $a_1, a_2, \cdots, a_n$ ($1 \leq a_i \leq 10^9$): the array values.

Output

Print the length of the longest increasing subsequence.

Examples

Standard Input	Standard Output
8 7 3 5 3 6 2 9 8	4

Problem D. Noldbach problem

Time limit:	please refer to DOM Judge
Memory limit:	please refer to DOM Judge

Nick is interested in prime numbers. Once he read about Goldbach problem. It states that every even integer greater than 2 can be expressed as the sum of two primes. That got Nick's attention and he decided to invent a problem of his own and call it Noldbach problem. Since Nick is interested only in prime numbers, he designed the Noldbach problem as follow: Please calculate how many prime numbers from 2 to N inclusively that can be expressed as the sum of three integer numbers: two neighboring prime numbers and 1. For example, $19 = 7 + 11 + 1$, or $13 = 5 + 7 + 1$.

Two prime numbers are called neighboring if there are no other prime numbers between them.

You are to help Nick, and find out the answer.

Input

The first line of the input contains an integers N.

- $2 \leq N \leq 10^6$

Output

Please output how many numbers from 2 to N inclusively that can be expressed as the sum of two neighboring prime numbers and 1.

Examples

Standard Input	Standard Output
30	2
5	0
7122	173

Problem E. Is this a TSP?

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

In a three-dimensional space, there are N cities: City $1, 2, \ldots, N$. City i is at point (X_i, Y_i, Z_i). No two cities share the same point. The cost it takes to travel from a city at point (a, b, c) to a city at point (p, q, r) is $|p - a| + |q - b| + \max(0, r - c)$. Find the minimum total cost it takes to start at City 1, visit all other cities at least once, and return to City 1.

Input

The first line contains an integer N ($2 \leq N \leq 17$).

After that are N lines, where the i-th line has three integers X_i, Y_i, Z_i ($-10^6 \leq X_i, Y_i, Z_i \leq 10^6$).

Output

Print the answer in one line.

Examples

Standard Input	Standard Output
2 0 0 0 1 2 3	9
3 0 0 0 1 1 1 -1 -1 -1	10
17 14142 13562 373095 -17320 508075 68877 223606 -79774 9979 -24494 -89742 783178 26457 513110 -64591 -282842 7124 -74619 31622 -77660 -168379 -33166 -24790 -3554 346410 16151 37755 -36055 51275 463989 37416 -573867 73941 -3872 -983346 207417 412310 56256 -17661 -42426 40687 -119285 43588 -989435 -40674 -447213 -59549 -99579 45825 7569 45584	6519344

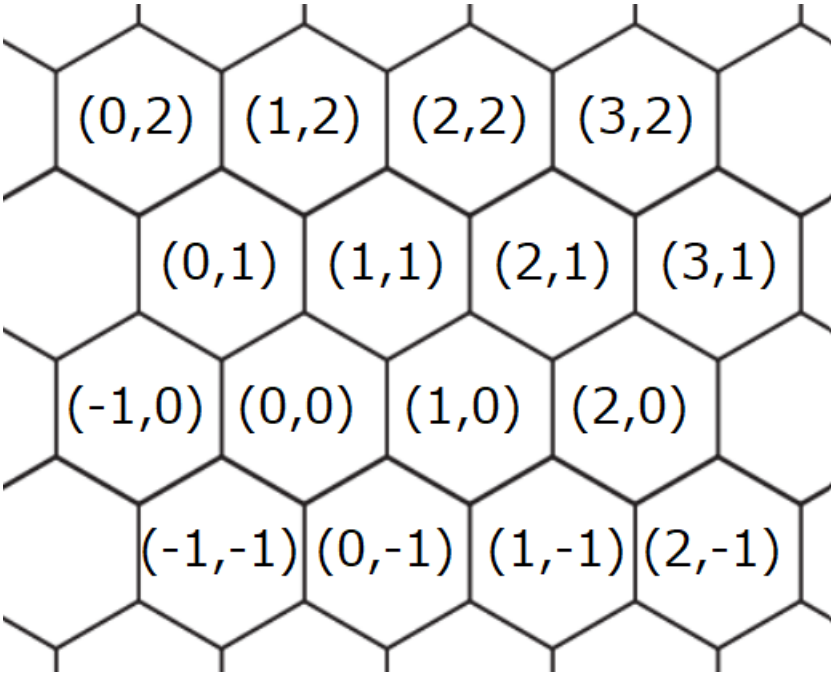
Note

For the first sample: The cost from City 1 to City 2 is $|1 - 0| + |2 - 0| + \max(0, 3 - 0) = 6$. The cost from City 2 to City 1 is $|0 - 1| + |0 - 2| + \max(0, 0 - 3) = 3$. Thus the total cost will be 9.

For the second sample: A possible way to obtain the minimum cost may be visiting the cities in the order $1, 2, 1, 3, 1$, which has total cost 10. Note that it is allowed to come back to City 1 on the way.

Problem F. Do use hexagon grid

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge



We have an infinite hexagonal grid shown above. Initially, all squares are white. A hexagonal cell is represented as (i, j) with two integers i and j. Cell (i, j) is adjacent to the following six cells:

- $(i-1, j-1)$
- $(i-1, j)$
- $(i, j-1)$
- $(i, j+1)$
- $(i+1, j)$
- $(i+1, j+1)$

Takahashi has painted N cells $(X_1, Y_1), (X_2, Y_2), ..., (X_N, Y_N)$ black.

Find the number of connected components formed by the black cells. Two black cells belong to the same connected component when one can travel between those two black cells by repeatedly moving to an adjacent black cell.

Input

First line contains an single integer N. Next N lines each contains two integer. The i-th line representing X_i, Y_i.

- $1 \leq N \leq 1000$
- $|X_i|, |Y_i| \leq 1000$
- The pairs (X_i, Y_i) are distinct.

Output

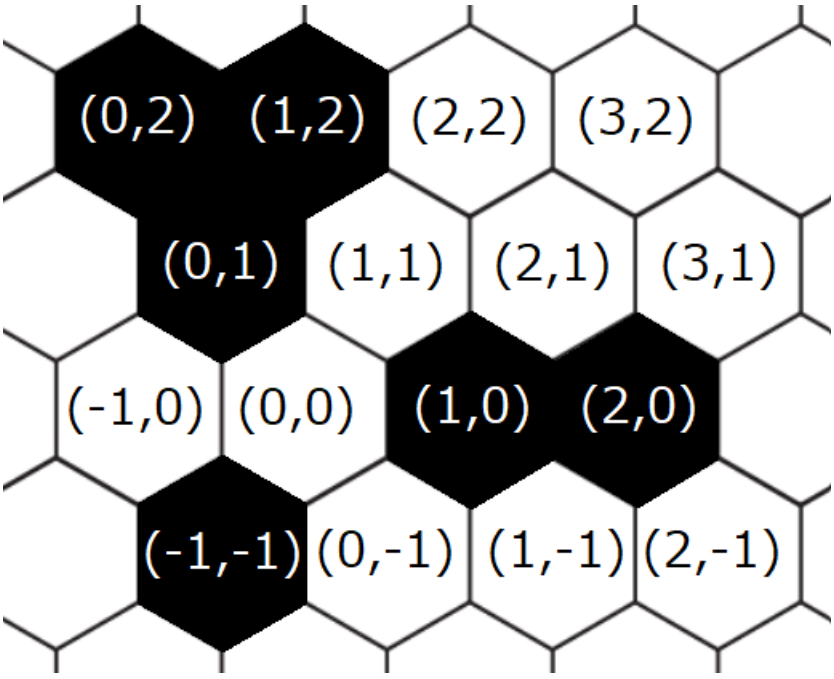
Output the number of connected components formed by the black cells.

Examples

Standard Input	Standard Output
6 -1 -1 0 1 0 2 1 0 1 2 2 0	3
4 5 0 4 1 -3 -4 -2 -5	4
5 2 1 2 -1 1 0 3 1 1 -1	1

Note

The hexagonal grid of the first example is as follow:



Problem G. Improving IT

Time limit: please refer to DOM Judge
Memory limit: please refer to DOM Judge

Your best friend is part of the business team at the Global Center for Parallel Computing (GCPC). She is responsible for buying and selling the hardware that is powering the system that will be in use for the next n months. Currently, she is planning the CPU replacement cycle for a single CPU. To ensure that the system is always up-to-date, the CPU must be replaced at least every m months. Fortunately, she can sell the replaced CPU to lower the overall costs to operate the new system. However, storage capacity is pricey, and she has to accept the resale value the CPU has in the month it is replaced. That means, when a CPU that was used for j months is replaced in month i, you need to sell the current CPU for the value it has after j months of usage and buy a new CPU for the price of the i-th month. She already compiled a list of CPU prices for the next n months including their resale value after 1 to m months. Note that you definitely need to buy a CPU in month 1 and you need to sell the last CPU in month $n+1$. How much money does the system cost at least over the n months?

Input

The input consists of:

- One line with two integers n and m ($1 \leq n, m$ and $nm \leq 5 \cdot 10^5$)
- n lines; the i-th line has an integer c ($0 \leq c \leq 10^9$), the cost of a CPU in month i, followed by $\min(m, n-i+1)$ integers c_j ($0 \leq c_j \leq 10^9$), the money you an earn by selling this CPU after $j > 0$ months.

Output

Output a single integer, the minimum total cost. Note that this number can be negative if reselling CPUs was profitable.

Examples

Standard Input	Standard Output
4 3 1000 900 800 900 700 600 500 400 1200 1200 1300 600 500	100
3 2 200 300 400 400 300 200 300 500	-400

Note

For the first sample: There are four months. The system requires a replacement at least every three months.

- In the first month, the CPU cost 1000. This CPU can be sold at 900 in the second month, 800 in the third month, 900 in the fourth month.
- In the second month, the CPU cost 700. This CPU can be sold at 600 in the third month, 500 in the fourth month, 400 in the fifth month.

- In the third month, the CPU cost 1200. This CPU can be sold at 1200 in the fourth month, 1300 in the fifth month.
- In the fourth month, the CPU cost 600. This CPU can be sold at 500 in the fifth month.

A strategy to obtain minimum cost 100 will be:

- In the first month, buy the CPU at price 1000.
- In the second month, sell the old one at price 900 and buy a new one at price 700.
- In the third month, sell the old one at price 600 and buy a new one at price 1200.
- In the fifth month, sell the old one at price 1300.

Problem H. Bit Inversions

Time limit: please refer to DOM Judg
Memory limit: please refer to DOM Judge

There is a bit string consisting of n bits. Then, there are some changes that invert one given bit. Your task is to report, after each change, the length of the longest substring whose each bit is the same. In other words, the length of the longest continuous bits that are same.

Input

The first input line has a bit string consisting of n bits. The bits are numbered $1, 2, ..., n$.

The next line contains an integer m: the number of changes.

The last line contains m integers $x_1, x_2, ..., x_m$ describing the changes.

- $1 \leq n \leq 2 \times 10^5$
- $1 \leq m \leq 2 \times 10^5$
- $1 \leq x_i \leq n$

Output

After each change, print the length of the longest substring whose each bit is the same.

Examples

Standard Input	Standard Output
001011 3 3 2 5	4 2 3

Note

In the example, the bit string first becomes 000011, then 010011, and finally 010001.