Problem A. Divisible Tree

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Today is Frikha's birthday , so we all decided to gift him an undirected tree , but not a usual tree , it's tree consisting of n nodes and $n-1$ edges , (a tree is a connected graph which contains no cycle and no loops) , the node 1 is the root of the tree and it's represented by $n-1$ integers which represetns the parents of the nodes 2 , 3 , ... and n (each node and his parent are connected by one edge , and for simplicity $parent[i] \leq i-1$ for all i from 2 to n).

A connected components of the tree , is a set S of nodes of the tree , such for every x , y in S , there exists p nodes in S , n_1 , n_2 ... n_p for some p such that there are edges between x-n_1 , n_1-n_2 .... n_p-y .

Let's suppose the i'th node in the tree has value a_i .

We call a connected components set S is divisble by x if for each node i in the set S we have a_i is divisible by x .

To have more fun in the birthday party , Mtaylor challenged Frikha to find the maximum size of a connected components divisible by some x $(2 \leq x)$, if there are many that have the same size you have to print minimum x possible and the size of the set .

Input

The first line contains one integer n $(1 \leq n \leq 3500\ )$.

The second line contains n integers a_i $(2 \leq a_i \leq 20000\ )$.

The third line contains $n-1$ integers p_i $(1 \leq p_i \leq i)$, the i'th integer represents the parents of the node $i+1$.

Output

Print two integers in one line , the minmum x which has the maximum size of connected components divisible by x and the size of such a set .

Examples

standard input	standard output
4 2 3 2 3 1 2 3	2 1
4 2 3 3 3 1 2 3	3 3

Problem B. Prizes

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

IPEIS CPC and IEEE Computer Society ENIS Student Chapter decided to buy n prizes for the winners , they have only two types of coins of values 1 and a , they have an infnite amount of every coin .

Given the prices of the n prizes , print the minimum number of coins needed to buy every prize in seperate lines.

Input

The first line contains two integers n and a $(1 \leq n\ ,\ a \leq 100000)$.

The second line contains n integers p_i $(1 \leq p_i \leq 100000)$, p_i is the price of the i'th prize .

Output

Print n lines , the i'th one contains the answer for the i'th prize.

Example

standard input	standard output
2 6 9 9	4 4

Problem C. Multiples Query

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Mtaylor is arithmetic and prefixes lover , Today it's his birthday , so his friends decided to give him an array a of n integers as a gift . He wants to play a game , every one of his friends will give him an integer r $(1 \le r \le n)$ and he needs to find the maximum length of subset S in the prefix subarray $[1, r]$ $(a[1], ..., a[r])$ containing distinct integers such that there is an integer x in S that divides all the numbers in the set S .

While Mtaylor was eating the cake , you decided to play the game instead of him , you are given q queries , the i'th one contains an integer r_i , and your task is to give the answer of the problem in the range $[1, r_i]$ in seperate lines .

Input

The first line contains 2 integers n and q $(1 \le q \le n \le 100000)$.

The second line contains n integers a_i the elemnts of the array $(1 \le a_i \le 10000\ )$.

The last line contains q integers r_i $(1 \le r_i \le n)$, where r_i represents the ranges of the i'th query .

Output

Print q lines , the answers to the queries .

Example

standard input	standard output
5 5 2 4 3 1 2 5 2 4 1 3	4 2 4 1 2

Problem D. Prime Query

Input file:	`standard input`
Output file:	`standard output`
Time limit:	3 seconds
Memory limit:	256 megabytes

While Mtaylor was learning how to check prime numbers , he got an idea of a problem to check prime numbers in a given array , so he asked you to help him .

You are given an array a of n integers a_i ($1 \leq a_i \leq 100000$), and q queries of 2 types :

Type 1: you are given two integers l and r ($l \leq r$) and you need to print the number of prime integers in the subarray from position l to r inclusive $(a[l], .., a[r])$.

Type 2: you are given two integers p and x and you should add x to the element of the array in position p.

Input

The first line contains 2 integers n and q ($1 \leq n$,$q \leq 100000$) .

The second line contains n integers a_i ($1 \leq a_i \leq 100000$) the elements of the array .

The next q lines contains the query inputs , the ith of them contains 3 integers t_i x_i,y_i ($1 \leq t_i \leq 2$).

If $t_i = 1$ then ($1 \leq x_i \leq y_i \leq n$), query of type 1 .

If $t_i = 2$ then ($1 \leq x_i \leq n$, $1 \leq y_i \leq 10$), query of type 2.

It's guranteed that the first type of query exists at least one time .

Output

Output the answers to the first type of queries in seperate lines .

Example

standard input	standard output
5 5 1 2 3 4 5 1 1 5 2 1 1 1 1 5 2 5 1 1 1 5	3 4 3

Problem E. Weird DNA

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

In some science competition , there is a challenge called Weird DNA , which consists of n sets of half chromosomes with distinct sizes of fype 2^i . The sets are represented by integers . Let's consider a set S represented by some integer x , there is a half chromosome in the set S if and only if the i'th bit is on in the binary representation of the integers x . For exemple , $x = 5$, $(5)_{(10)} = (101)_{(2)}$, then in the set there is one half chromosome of size 2^0 and one half chromosome of size 2^2.

Now after understanding what the game consists of , the challenge is to find some consecutive sets which can create a good DNA , a good DNA is where we can split the half chromosomes in pairs such that every pair contains 2 half chromosome of the same size .

You consider this challenge so easy , so instead of finding such sets , you will calculate the number of ways to choose some consecutive sets that forms a good DNA .

Input

The first line contains one integer n ($1 \leq n \leq 100000$).

The second line contains n integers a_i ($1 \leq a_i \leq 1000000$) the i'th integer is the representative of the i'th set.

Output

Output one integer , the answer to the problem .

Example

standard input	standard output
4 1 1 1 1	4

Note

In the first example : the consecutive sets possibles are $(1, 2)$, $(2, 3)$, $(3, 4)$ and $(1, 2, 3, 4)$, only 4 possibilites , so the answer equals 4.

Problem F. Grid Coloring

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Mtaylor has a multicolored chessboard with n rows and m colums, but he doesn't like such a board with so many colors , so he decided to color some of it's cells to make every row in the chessboard has exactly 2 distinct colors and the whole chessboard has exactly 2 distinct colors .

Let's suppose the color of the i'th row and j'th column is $x_{i,j}$, and all the colors that Mtaylor has are numbered from 1 to k and the colors in the chessboard are between 1 and k

Unfortunately Mtaylor is busy , so he asked you to help him to find the minimum number of cells to color , such that the chessboard will fulfill his conditions .

Input

The first line contains 3 integers n , m and k ($2 \leq n$, $m \leq 1000$,$2 \leq k \leq 1000000$).

The next n lines each contains m integers $x_{i,j}$ $(1 \leq x_{i,j} \leq k)$ where $x_{i,j}$ the color of the cell in the ith row and jth column .

Output

Print one integer , the answer of the problem .

Example

standard input	standard output
2 2 3 1 2 2 1	0

Problem G. Winnable Game

Input file:	`standard input`
Output file:	`standard output`
Time limit:	2 seconds
Memory limit:	256 megabytes

Fares and Omar are best friends , one day they got bored so they invented a new game , the new games consists of $n+2$ points , the first point is $x = 0$ and the $(n+2)$'th point is $x = n+1$, the points from 1 to n each contains one number a_i. Omar starts at $x = 0$ and fares starts at $x = n+1$. At each step , omar can move from point x to $x+1$ or fares can move from point x to $x-1$ (they can't both move at the same step).

Let's suppose $f(l, r) = a[l+1] + ... + a[r-1]$.

Let's suppose after some steps , omar is at position l and fares is at position r , they can win the game at this state only if $f(l, r) \leq k$ (k is fixed at the begining of the game).

Given the numbers written in the points from 1 to n and the number k , determine the minimum number of steps needed to win the game .

Input

The first line contains two integers n and k ($1 \leq n \leq 1000000$, $1 \leq k \leq 10000000$).

The second line contains n integers a_i ($1 \leq a_i \leq k$) the numbers written in the points 1 to n .

Output

Print one integer the minmum number of steps to win the game .

Examples

standard input	standard output
5 10 1 3 1 2 1	0
5 10 1 3 10 2 1	3

Problem H. Crypting "helloworld"

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1 second
Memory limit:	256 megabytes

Hello World contest is finally here , Farouk want to trick some of the IEEE Computer Society ENIS Student Chapter members , so he decided to crypt the word "*helloworld*" by the following algorithm:

let's suppose $pos_a = 0$, $pos_b = 1$, ... $pos_z = 25$, he choose a number a and for every character c in the word "*helloworld*" he replaces it by the charatcter which it's position = $(a \cdot pos_c)$ $mod(26)$ (where mod is the modulo operator that finds the remainder after division of one number by another).

He gave the crypted string to them and the number a , determine if they can decrypt the world to get "*helloworld*" or not (that means for every character in the given string they will be able to know the exact character that corrosponds to it).

Input

The input contains one integer a $(0 \leq a \leq 25)$.

Output

Print "YES" without quotes if they can decrypt the word to get "*helloworld*", otherwise print "NO" whithou quotes .

Examples

standard input	standard output
1	YES
2	NO

Problem I. Kill or Skip

Input file:	`standard input`
Output file:	`standard output`
Time limit:	1.5 seconds
Memory limit:	256 megabytes

After a long day of problem solving , Mtaylor decided to play a new game called Kill or Skip , the game consists of n monsters numbered from 1 to n , the games starts with the monster 1 then 2 .. and so on to n (you can't change the order) , at each level from 1 to n you can enter the room and kill the monster or skip the room and go to the next room .

But the game isn't that easy , when you enter the i'th room you get hit first by dmg_i from the monster and your health will decrease by dmg_i (if your health reaches 0 or below your charatcter will die), then you can start shooting it , you need $ammo_i$ to kill it (if you ran out of ammo and the monster isn't dead , then your character will die) , but if you killed the monster you will be rewarded xp_i xp points , hp_i health points and ra_i additional ammo . You start the game with 100 health points and 100 ammo and this is the maximum possible that you can reach , that means if your health is 98 and you get 5 additional health points , your health points will be 100 and not 103 , same thing for the ammo .

But we know the game isn't fun without a magical spell , there is only one magical spell and you can use it only once (in only one room of your choice) , the magical spell square the xp points gained after killing a monster that mean instead of gaining xp you will get xp^2 .

Mtaylor is tired , so he asked you to help him to find a srategie to make the maximum xp points possible knowing that you start the game with 0 xp . Print that maximum score .

Input

The first line contains one integer n ($1 \leq n \leq 100$).

Then n lines , the i'th one contains 5 integers dmg_i , $ammo_i$, xp_i , hp_i and ra_i ($1 \leq dmg_i$, $ammo_i \leq 100$, $0 \leq xp_i \leq 10000$, $0 \leq hp_i$, $ra_i \leq 100$).

Output

Print one integer the answer to the problem .

Examples

standard input	standard output
2 50 70 10 20 30 50 50 20 0 0	410
2 50 70 10 20 19 50 50 20 0 0	400

Problem J. Another Tree Problem

Input file:	`standard input`
Output file:	`standard output`
Time limit:	5 seconds
Memory limit:	256 megabytes

Mtaylor has a new tree , but it's not like all trees . It's a rooted tree .

Let's call $d(x)$ as the depth node x . Depth of the root if 1 and is x is the parent of y then $d(y)=d(x)+1$.

The tree is defined by the number of children of nodes , that means each node of depth x has a_x children . Maximum possible depth of a node is n, and $a_n = 0$.

We define $path(k)$ as the number of unordered pairs of vertices in the tree such that the number of edges on the simple path between them is equal to k.

Calculate $path(k)$ modulo $10^9 + 7$ for every $1 \leq k \leq 2n - 2$.

Input

The first line of input contains an integer n ($2 \leq n \leq 5000$) — the maximum depth of a node.

The second line of input contains $n - 1$ integers a_1, a_2, ..., a_{n-1} ($2 \leq a_i \leq 10^9$), where a_i is the number of children of every node x such that $d(x) = i$.

Output

Print $2n - 2$ numbers. The k-th of these numbers must be equal to $path(k)$ modulo $10^9 + 7$.

Example

standard input	standard output
3 2 3	8 13 6 9