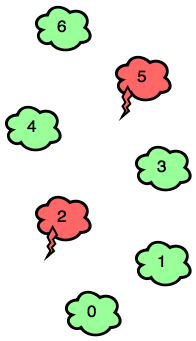
**Solution 1:**

Emma is playing a new mobile game involving n clouds numbered from 0 to n−1. A player initially starts out on cloud c0c0, and they must jump to cloud cn−1. In each step, she can jump from any cloud ii to cloud i+1 or cloud i+2.

There are two types of clouds, *ordinary clouds* and *thunderclouds*. The game ends if Emma jumps onto a thundercloud, but if she reaches the last cloud (i.e., cn−1), she wins the game!



Can you find the minimum number of jumps Emma must make to win the game? It is guaranteed that clouds c0and cn−1 are ordinary-clouds and it is *always possible* to win the game.

**Input Format**

The first line contains an integer, n (the total number of clouds).   
The second line contains n space-separated binary integers describing clouds c0,c1,…,cn−1

* If ci=0, the ith cloud is an ordinary cloud.
* If ci=1, the ith cloud is a thundercloud.

**Constraints**

* 2≤n≤100
* ci∈{0,1}
* c0=cn−1=0

**Output Format**

Print the minimum number of jumps needed to win the game.

**Sample Input 0**

7

0 0 1 0 0 1 0

**Sample Output 0**

4

**Sample Input 1**

6

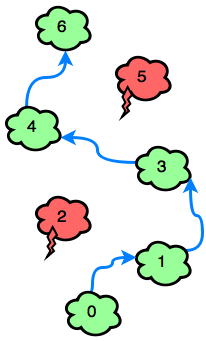
0 0 0 0 1 0

**Sample Output 1**

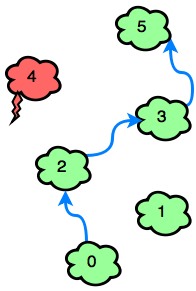
3

**Explanation**

*Sample Case 0:*   
Because c2 and c5 in our input are both 1, Emma must avoid c2 and c5. Bearing this in mind, she can win the game with a minimum of 4 jumps:



*Sample Case 1:*   
The only thundercloud to avoid is c4. Emma can win the game in 3 jumps:



Solution 2

Erica wrote an increasing sequence of nn numbers (a0,a1,…,an−1) in her notebook. She considers a triplet (ai,aj,ak) to be beautiful if:

* i<j<k
* a[j]−a[i]=a[k]−a[j]=d

Given the sequence and the value of d, can you help Erica count the number of beautiful triplets in the sequence?

**Input Format**

The first line contains 2 space-separated integers, n (the length of the sequence) and d (the beautiful difference), respectively.   
The second line contains n space-separated integers describing Erica's increasing sequence, a0,a1,…,an−1.

**Constraints**

* 1≤n≤104
* 1≤d≤20
* 0≤ai≤2×104
* ai>ai−1 for 0<i≤n−1

**Output Format**

Print a single line denoting the number of beautiful triplets in the sequence.

**Sample Input**

7 3

1 2 4 5 7 8 10

**Sample Output**

3

**Explanation**

Our input sequence is 1,2,4,5,7,8,10 and our beautiful difference d=3. There are many possible triplets (ai,aj,ak), but our only beautiful triplets are (1,4,7), (4,7,10)and (2,5,8). Please see the equations below:

7−4=4−1=3=d  
10−7=7−4=3=d  
8−5=5−2=3=d

Recall that a beautiful triplet satisfies the following equivalence relation: a[j]−a[i]=a[k]−a[j]=d where i<j<k.

Solution3

The [Tower of Hanoi](https://en.wikipedia.org/wiki/Tower_of_Hanoi) is a famous game consisting of 3 rods and a number of discs of incrementally different diameters. The puzzle starts with the discs neatly stacked on one rod, ordered by ascending size with the smallest disc at the top. The game's objective is to move the entire stack to another rod, obeying the following rules:

1. Only one disc can be moved at a time.
2. Each move consists of taking the topmost disc from a stack and moving it to the top of another stack.
3. No disc may be placed on top of a *smaller* disc.

Gena has a modified version of the [Tower of Hanoi](https://en.wikipedia.org/wiki/Tower_of_Hanoi). His *Hanoi* has 44 rods and NN discs ordered by ascending size. He made a few moves (following the rules above), but stopped and lost his place. He wants to restore the tower to its original state by making valid moves. Given the state of Gena's *Hanoi*, help him calculate the minimum number of moves needed to restore the tower to its original state.

**Note:** Gena's rods are numbered from 11 to 44. All discs are initially located on rod 11.

**Input Format**

The first line contains a single integer, NN, denoting the number of discs.   
The second line contains NN space-separated integers, where the ithith integer is the index of the rod where the disk with diameter ii is located.

**Constraints**

* 1≤N≤101≤N≤10

**Output Format**

Print the minimum number of moves Gena must make to restore the tower to its initial, ordered state on the first rod.

**Sample Input**

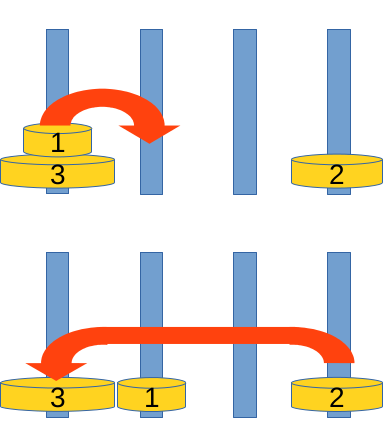
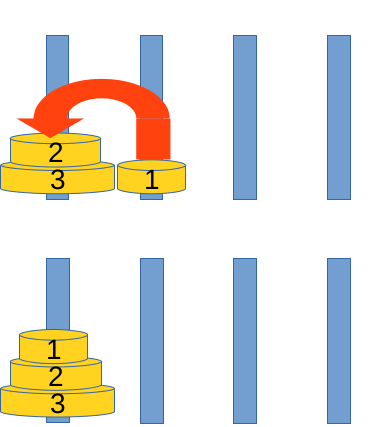
3

1 4 1

**Sample Output**

3

**Explanation**

33 moves are enough to build the tower. Here is one possible solution:   
  


Solution 4

This challenge uses the famous [KMP algorithm](https://en.wikipedia.org/wiki/Knuth%E2%80%93Morris%E2%80%93Pratt_algorithm). It isn't really important to understand how KMP works, but you should understand what it calculates.

A KMP algorithm takes a string, SS, of length NN as input. Let's assume that the characters in SS are indexed from 11 toNN; for every prefix of SS, the algorithm calculates the length of its longest valid [border](http://algorithmsforcontests.blogspot.com/2012/08/borders-of-string.html) in linear complexity. In other words, for every ii (where 1≤i≤N1≤i≤N) it calculates the largest ll (where 0≤l≤i−10≤l≤i−1) such that for every pp (where 1≤p≤l1≤p≤l) there is S[p]=S[i−l+p]S[p]=S[i−l+p].

Here is an implementation example of KMP:

kmp[1] = 0;

for (i = 2; i <= N; i = i + 1){

l = kmp[i - 1];

while (l > 0 && S[i] != S[l + 1]){

l = kmp[l];

}

if (S[i] == S[l + 1]){

kmp[i] = l + 1;

}

else{

kmp[i] = 0;

}

}

Given a sequence x1,x2,…,x26x1,x2,…,x26, construct a string, SS, that meets the following conditions:

1. The frequency of letter 'aa' in SS is exactly x1x1, the frequency of letter 'bb' in SS is exactly x2x2, and so on.
2. Let's assume characters of SS are numbered from 11 to NN, where ∑i=1nxi=N∑i=1nxi=N. We apply the KMP algorithm to SS and get a table, kmpkmp, of size NN. You must ensure that the sum of kmp[i]kmp[i] for all ii is minimal.

If there are multiple strings which fulfill the above conditions, print the [lexicographically](https://en.wikipedia.org/wiki/Lexicographical_order) smallest one.

**Input Format**

A single line containing 2626 space-separated integers describing sequence xx.

**Constraints**

* The sum of all xixi will be a positive integer ≤106≤106.

**Output Format**

Print a single string denoting SS.

**Sample Input**

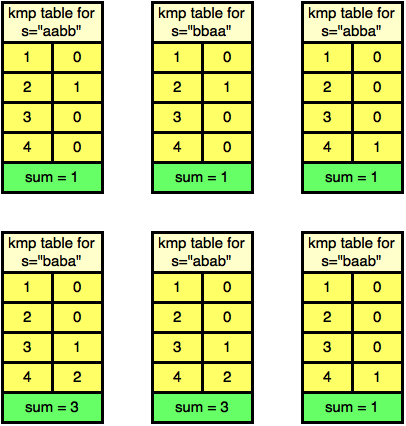
2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

**Sample Output**

aabb

**Explanation**

The output string must have two 'aa' and two 'bb'. There are several such strings but we must ensure that sum of kmp[i]kmp[i] for all 1<=i<=41<=i<=4 is minimal. See the figure below:



The minimum sum is 11. Among all the strings that satisfy both the condition, "aabb" is the lexicographically smallest.

Solution 5

Alice and Bob are playing a game, defined below:

* There is an undirected tree graph with nn nodes that has the following properties:
  + Each node has cici golden coins.
  + Node 11 is root of the tree.
  + The parent node of some node uu is defined as p(u)p(u).
* Moves
  + Players move in turns.
  + During a move, a player can select a node u>1u>1 and move one or more coins to p(u)p(u).
  + If the current player can't make any move, they lose the game.

The game quickly becomes boring because the result is determined by the tree's configuration and the number of coins in each node (assuming that both players play optimally).

Alice decides to instead challenge Bob by asking him qq questions. For each question ii:

1. Alice picks a node ui>1ui>1 and *removes* the edge between uiui and p(ui)p(ui).
2. She picks another node vv and draws a new undirected edge between uiui and vivi. So now p(ui)=vip(ui)=vi.

Bob must determine if the first player has a winning strategy for the new tree or not. It's possible that after Alice draws the new edge, the graph will no longer be a tree; if that happens, the question is *invalid*. Each question is independent, so the answer depends on the initial state of the graph (and not on previous questions).

Given the tree and the number of coins in each node, can you help Bob answer all qq questions?

**Input Format**

The first line contains an integer, nn (the number of nodes).   
The second line contains nn space-separated integers, c1,c2,…,cnc1,c2,…,cn, describing the number of coins in each node.   
Each of the n−1n−1 subsequent lines contains 22 space-separated integers denoting an undirected edge between nodes aa and bb, respectively.   
The next line contains an integer, qq (the number of questions Alice asks).   
Each of the qq subsequent lines contains 22 space-separated integers, uiui and vivi, respectively.

**Constraints**

* 1≤n,q≤5×1041≤n,q≤5×104
* 1≤a,b≤n1≤a,b≤n
* 0≤ci≤200≤ci≤20

For each question:

* 2≤ui≤n2≤ui≤n
* 1≤vi≤n1≤vi≤n
* ui≠viui≠vi

**Output Format**

On a new line for each question, print YESYES if the first player has a winning strategy, print NONO if they do not, or print INVALIDINVALID if the question is not valid.

**Sample Input**

6

0 2 2 1 3 2

1 2

1 3

3 4

3 5

4 6

3

6 2

4 1

3 6

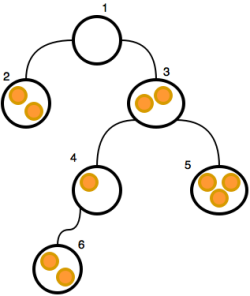
**Sample Output**

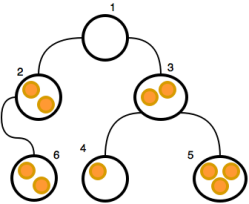
NO

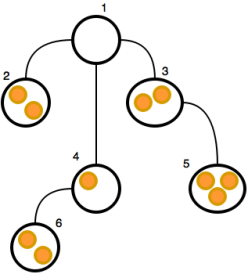
YES

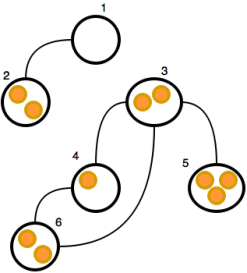
INVALID

**Explanation**

Initally the tree looks like this:

After the first question (6 26 2), the tree looks like this:Alice removes the edge conecting node 66 to 44 and makes 22 the new parent node of 66. Because this configuration does not result in a winning strategy, we print NONO on a new line.

After the second question (4 14 1), the tree looks like this:Alice removes the edge conecting node 44 to 33 and makes 11 the new parent node of 44. Because this configuration results in a winning strategy, we print YESYES on a new line.

After the third question (3 63 6), the graph is no longer a tree:Alice removes the edge conecting node 33 to 11 and makes 66 the new parent node of 33. The graph is now partitioned into two separate subgraphs (one of which is also not a tree); because the game must be played on a single undirected tree graph, we print INVALIDINVALID on a new line.

s

Solution 6

Daniel loves graphs. He thinks a graph is *special* if it has the following properties:

* It is undirected.
* The length of each edge is 11.
* It includes *exactly* PP different *lovely triplets*.

A *triplet* is a set of 33 different nodes. A triplet is *lovely* if the minimum distance between each pair of nodes in the triplet is *exactly* QQ. Two triplets are different if 11 or more of their component nodes are different.

Given PP and QQ, help Daniel draw a *special graph*.

**Input Format**

A single line containing 22 space-separated integers, PP (the number of different lovely triplets you must have in your graph) and QQ (the required *distance* between each pair of nodes in a lovely triplet), respectively.

**Constraints**

* 1≤P≤50001≤P≤5000
* 2≤Q≤92≤Q≤9

**Output Format**

For the first line, print 22 space-separated integers, NN (the number of nodes in the graph) and MM (the number of edges in the graph), respectively.   
On each line ii of the MM subsequent lines, print two space-separated integers, uiui and vivi, describing an edge between nodes uiui and vivi.

Your output must satisfy the following conditions:

* 0≤N,M≤1000≤N,M≤100
* 1≤ui,vi≤N1≤ui,vi≤N

If there is more than one correct answer, print any one of them.

**Sample Input**

3 2

**Sample Output**

7 7

1 2

2 3

3 4

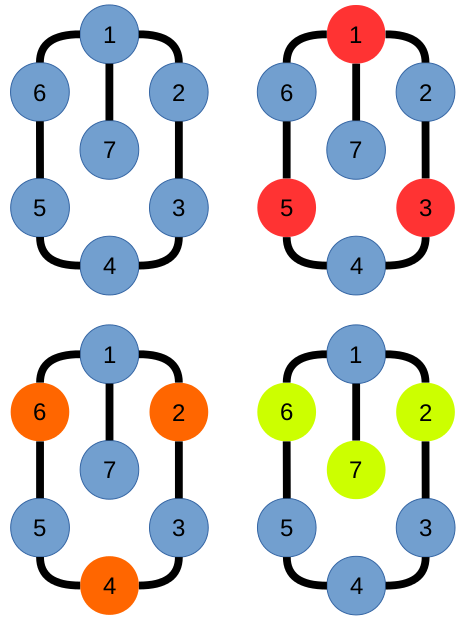
4 5

5 6

6 1

1 7

**Explanation**

There are exactly P=3P=3 lovely triplets in this graph: {1,3,5}{1,3,5}, {2,4,6}{2,4,6}, and {2,6,7}{2,6,7}.Observe that each node in a lovely triplet is Q=2Q=2 edges away from the other nodes composing the lovely triplet.

Solution 7

Alexey is playing with an array, AA, of nn integers. His friend, Ivan, asks him to calculate the sum of the maximum values for all subsegments of AA. More formally, he wants Alexey to find F(A)=∑l=1n∑r=ln maxl≤x≤r A[x]F(A)=∑l=1n∑r=ln maxl≤x≤r A[x].

Alexey solved Ivan's challenge faster than expected, so Ivan decides to add another layer of difficulty by having Alexey answer mm queries. The ithith query contains subsegment [Li,Ri][Li,Ri], and he must calculate the sum of maximum values on all subsegments inside subsegment [Li,Ri][Li,Ri].

More formally, for each query ii, Alexey must calculate the following function:

F(A,Li,Ri)=∑l=LiRi∑r=lRimaxl≤x≤r A[x]F(A,Li,Ri)=∑l=LiRi∑r=lRimaxl≤x≤r A[x].

Can you help Alexey solve this problem?

**Input Format**

The first line contains 22 space-separated positive integers, nn (the length of array AA) and mm (number of queries), respectively.   
The second line contains nn space-separated integers, a0,a1,…,an−1a0,a1,…,an−1 describing each element ajaj (where 0≤j<n0≤j<n) in array AA.   
Each of the mm subsequent lines contains 22 space-separated positive integers describing the respective values for LiLiand RiRi in query ii (where 0≤i<m0≤i<m).

**Constraints**

* 1≤n,m≤1350001≤n,m≤135000
* −109≤ai≤109−109≤ai≤109
* 1≤Li≤Ri≤n1≤Li≤Ri≤n

**Output Format**

For each query ii (where 0≤i<m0≤i<m), print its answer on a new line.

**Sample Input**

3 6

1 3 2

1 1

1 2

1 3

2 2

2 3

3 3

**Sample Output**

1

7

15

3

8

2

**Explanation**

The answer for the second query is shown below: F(A,1,2)=max1≤x≤1A[x]+max1≤x≤2A[x]+max2≤x≤2A[x]F(A,1,2)=max1≤x≤1A[x]+max1≤x≤2A[x]+max2≤x≤2A[x] =1+3+3=7=1+3+3=7

The answer for the third query is shown below: F(A,1,3)=max1≤x≤1A[x]+max1≤x≤2A[x]+max1≤x≤3A[x]+max2≤x≤2A[x]+max2≤x≤3A[x]+max3≤x≤3A[x]F(A,1,3)=max1≤x≤1A[x]+max1≤x≤2A[x]+max1≤x≤3A[x]+max2≤x≤2A[x]+max2≤x≤3A[x]+max3≤x≤3A[x] =1+3+3+3+3+2=15

Solution 8

Shashank loves trees and math. He has a rooted tree, TT, consisting of NN nodes uniquely labeled with integers in the inclusive range [1,N][1,N]. The node labeled as 11 is the *root* node of tree TT, and each node in TT is associated with some positive integer value (all values are initially 00).

Let's define FkFk as the kthkth [Fibonacci number](https://en.wikipedia.org/wiki/Fibonacci_number). Shashank wants to perform 22 types of operations over his tree, TT:

1. UU XX kk  
   Update the subtree rooted at node XX such that the node at level 00 in subtree XX (i.e., node XX) will have FkFkadded to it, all the nodes at level 11 will have Fk+1Fk+1 added to them, and so on. More formally, all the nodes at a distance DD from node XX in the subtree of node XX will have the (k+D)th(k+D)th Fibonacci number added to them.
2. QQ XX YY  
   Find the sum of all values associated with the nodes on the unique path from XX to YY. Print your sum modulo 109+7109+7 on a new line.

Given the configuration for tree TT and a list of MM operations, perform all the operations efficiently.

**Note:** F1=F2=1F1=F2=1.

**Input Format**

The first line contains 22 space-separated integers, NN (the number of nodes in tree TT) and MM (the number of operations to be processed), respectively.   
Each line ii of the N−1N−1 subsequent lines contains an integer, PP, denoting the parent of the (i+1)th(i+1)th node.   
Each of the MM subsequent lines contains one of the two types of operations mentioned in the *Problem Statement*above.

**Constraints**

* 1≤N,M≤1051≤N,M≤105
* 1≤X,Y≤N1≤X,Y≤N
* 1≤k≤10151≤k≤1015

**Output Format**

For each operation of type 22 (i.e., QQ), print the required answer modulo 109+7109+7 on a new line.

**Sample Input**

5 10

1

1

2

2

Q 1 5

U 1 1

Q 1 1

Q 1 2

Q 1 3

Q 1 4

Q 1 5

U 2 2

Q 2 3

Q 4 5

**Sample Output**

0

1

2

2

4

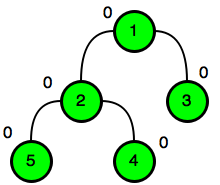
4

4

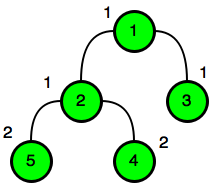
10

**Explanation**

Intially, the tree looks like this:



After update operation 1 11 1, it looks like this:



After update operation 2 22 2, it looks like this:

