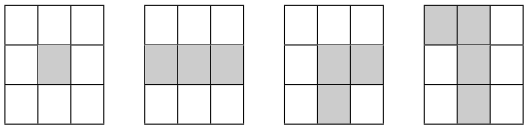
Task 1: `L-Board`

Lord Pooty has a n by m board of integers A and would like to draw an L. However, he would like to maximise the sum of integers on the tiles covered by L. The L can be rotated in all 4 possible orientations such that the sides are parallel to the board. Each side of the L may not necessarily be drawn (a straight line is possible). Some examples of valid Ls are shown below:



Formally, you want to choose 3 points, (x_1, y_1), (x_2, y_1) and (x_1, y_2) (which may not necessarily be distinct) on the board A such that

$$V = \sum_{i=\min(x_1,x_2)}^{\max(x_1,x_2)} A_{i,y_1} + \sum_{j=\min(y_1,y_2)}^{\max(y_1,y_2)} A_{x_1,j} - A_{x_1,y_1}$$

is maximised.

Input format

Your program must read from standard input.

The input starts with a line with two integers n and m where n and m are height and width of the board. This is followed by n lines of m integers, representing the board.

Output format

Your program must print to standard output.

The output should contain a single integer on a single line, the maximum V possible.

Subtasks

The maximum execution time on each instance is 2.0s, and the maximum memory usage on each instance is 1GB. For all testcases, the input will satisfy the following bounds:

- $1 \leq n, m \leq 1000$
- $-10^9 \leq A_{i,j} \leq 10^9$ for $1 \leq i \leq n$ and $1 \leq j \leq m$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	5	$1 \leq n, m \leq 2$
2	10	$n = 1$
3	15	$1 \leq n, m \leq 100$
4	15	$1 \leq n, m \leq 300$
5	25	$0 \leq A_{i,j} \leq 10^9$ for $1 \leq i \leq n$ and $1 \leq j \leq m$
6	30	No additional restrictions

Sample Testcase 1

This testcase is valid for subtasks 1, 3, 4, 5 and 6.

Input	Output
2 2 8 1 3 4	15

Sample Testcase 1 Explanation

In this example, you choose 8, 3, 4 to form an L.

Sample Testcase 2

This testcase is valid for subtasks 2, 3, 4, and 6.

Input	Output
1 8 -2 -1 8 -2 9 0 -2 1	15

Sample Testcase 2 Explanation

You draw a line covering 8, -2 and 9.

Task 2: `Tree Cutting`

A country has N cities numbered from 1 to N and $N-1$ bidirectional highways. It is possible to travel from any city to any other city using only the highways.

The *distance* between two cities x and y is defined as the number of highways required to travel from x to y.

The governor has decided to demolish a highway and build another highway such that the largest distance between any two cities is maximized.

Find this maximum largest distance.

Input

Your Program must read from standard input.

The first line contains an integer, N, the number of cities.

In the next $N-1$ lines, each line contains 2 distinct integers u and v, representing a highway connecting cities u and v.

Output

Your program must print to standard output.

The output should contain a single integer on a single line, the new largest distance between any two cities.

Subtasks

The maximum execution time on each instance is 3.0s, and the maximum memory usage on each instance is 1GB. For all testcases, the input will satisfy the following bounds:

- $3 \leq N \leq 300\,000$
- $1 \leq u, v \leq N$

Your program will be tested on input instances that satisfy the following restrictions:

Subtask	Marks	Additional Constraints
1	5	$N \leq 10$
2	10	$N \leq 100$
3	15	$N \leq 3000$
4	15	$N \leq 300\,000$, there is at most one city with at least 3 highways connected to it.
5	55	-

Sample Testcase 1

This testcase is valid for all subtasks.

Input	Output
4 1 2 1 3 3 4	3

Sample Testcase 1 Explanation

It is impossible to increase the largest distance beyond 3.

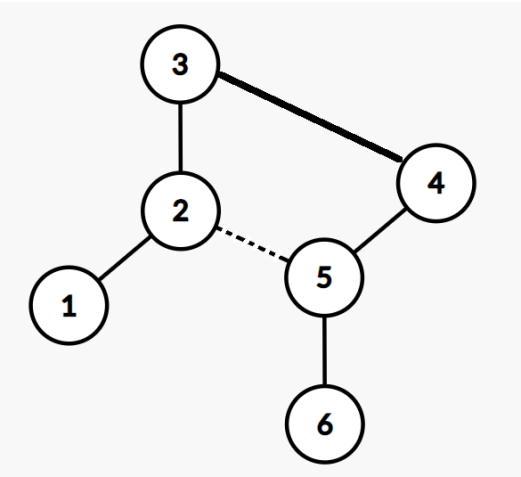
Sample Testcase 2

This testcase is valid for subtasks $1, 2, 3$ and 5.

Input	Output
6 1 2 2 3 2 5 4 5 5 6	5

Sample Testcase 2 Explanation

We may remove highway 2-5, and add the highway 3-4, so that the longest path becomes $1 - 2 - 3 - 4 - 5 - 6$.



Task 3: `Dragonfly`

Dragonflies can be seen around ponds at Botanic Gardens and Bishan Park. In one of the denser forested areas, Benson the Rabbit has noted down n ponds that the dragonflies fly around. At pond i ($1 \leq i \leq n$), there are $b[i]$ bugs that the dragonflies can eat. The bugs at pond i belong to species $s[i]$.

Benson has also noted down $n-1$ trails. Each trail j ($1 \leq j < n$) connects 2 distinct ponds $u[j]$ and $v[j]$ bidirectionally. Dragonflies can travel from any pond to any other pond using only the trails.

Benson has captured d dragonflies and intends to release them one at a time at pond 1. Dragonfly k ($1 \leq k \leq d$) has a home pond of $h[k] \neq 1$ and will travel to pond $h[k]$ without visiting any pond more than once using only the trails. These dragonflies will be released in increasing order from dragonfly 1 to dragonfly d. After a dragonfly is released, it will eat a single bug (if there is one or more bugs remaining) at every pond that it visits (including pond 1), reducing the number of bugs at each of those ponds by 1 if it is not 0.

Help Benson determine the number of **distinct species** of bugs eaten during the journey of each of the d dragonflies.

Input format

The input format is as follows:

- The first line of input contains 2 spaced integers n and d respectively.
- The next line of input contains n spaced integers $b[1], b[2], \cdots, b[n]$.
- The next line of input contains n spaced integers $s[1], s[2], \cdots, s[n]$.
- The next line of input contains d spaced integers $h[1], h[2], \cdots, h[d]$.
- The next $n-1$ lines of input contains 2 spaced integers each. The i^{th} of these lines contains $u[i]$ and $v[i]$ respectively.

Output format

Output a single line with d spaced integers. The k^{th} of these integers should be the number of distinct species of bugs eaten by the k^{th} dragonfly.

Subtasks

The maximum execution time on each instance is 5.0s, and the maximum memory usage on each instance is 1GB. For all testcases, the input will satisfy the following bounds:

- $2 \leq n \leq 2 \cdot 10^5$
- $1 \leq d \leq 2 \cdot 10^6$
- $1 \leq s[i] \leq n$ (for all $1 \leq i \leq n$)
- $0 \leq b[i] \leq d$ (for each $1 \leq i \leq n$)
- $1 \leq u[j], v[j] < n, u[j] \neq v[j]$ (for each $1 \leq j \leq n-1$)
- $2 \leq h[k] \leq n$ (for each $1 \leq k \leq n$)

Subtask	Marks	Additional Constraints
1	10	$n, d \leq 1000$
2	10	$d \leq 2 \cdot 10^5, b[i] = d$ (for all $1 \leq i \leq n$)
3	12	$d \leq 2 \cdot 10^5, b[i] \leq 10$ (for all $1 \leq i \leq n$)
4	12	$d \leq 2 \cdot 10^5, u[j] = j, v[j] = j+1$ (for all $1 \leq j \leq n-1$)
5	37	$d \leq 2 \cdot 10^5, s[i] = i$ (for all $1 \leq i \leq n$)
6	16	$d \leq 2 \cdot 10^5$
7	3	No additional restrictions

Sample Testcase 1

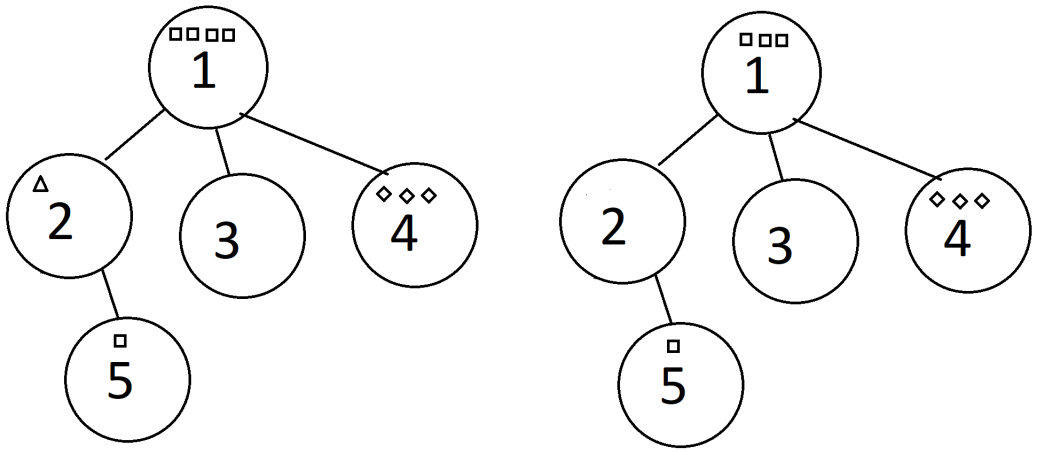
This testcase is valid for subtasks 1, 3, 6 and 7.

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

Sample Testcase 1 Explanation

The input can be visualized using the below figure where □ represents a bug of species 1, △ represents a bug of species 2, and ◇ represents a bug of species 3.

Initially, the bug configuration matches that of the left diagram. When the first dragonfly flies to pond 2, 1 bug each of species 1 and 2 is eaten. The configuration changes to the right diagram.



After that, a dragonfly flies to pond 5. It eats a bug from pond 1 and a bug from pond 5. Note that it does not eat any bugs from pond 2 because there are no more bugs left in pond 2. Although 2 bugs are eaten in total, they are of the same species □, so only 1 distinct species of bugs is eaten.

Sample Testcase 2

This testcase is valid for subtasks 1, 3, 6 and 7.

Input	Output
7 4 0 2 4 4 0 1 3 6 1 6 2 2 2 1 7 5 2 4 4 1 4 5 6 2 1 6 1 3 6 7	2 1 1 1