ALGORITHMS

Lab 01Tasks

A. Odd or Even?

Time limit: 1 second
Memory limit: 256 MB

You are given T numbers, and for each number, you have to tell whether it's odd or even.

Input:
- First line: Integer T (1 ≤ T ≤ 100)
- Next T lines: Each line contains an integer N (−10^5 ≤ N ≤ 10^5)

Output:
- For each number, print "{N} is an Even number." or "{N} is an Odd number."

Example:
Input:
5
10
19
7
3
100

Output:
10 is an Even number.
19 is an Odd number.
7 is an Odd number.
3 is an Odd number.
100 is an Even number.

B. Can you solve Arithmetic Expressions?

Time limit: 1 second
Memory limit: 256 MB

Solve arithmetic expressions and print the result.

Input:
- First line: T (1 ≤ T ≤ 1000)
- Next T lines: Each line contains an expression like "calculate 67 + 41"

Output:
- Print the result with 6 decimal places

Example:
Input:
3
calculate 67 + 41
calculate 85 / 5
calculate 13 - 56

Output:
108.000000
17.000000
-43.000000

C. Array Reverse

Time limit: 1 second
Memory limit: 256 MB

Reverse the array and print the last K elements.

Input:
- First line: Two integers N and K (1 ≤ K ≤ N ≤ 10^6)
- Second line: N space-separated integers

Output:
- K space-separated integers from reversed array

Example:
Input:
5 3
5 6 7 8 9

Output:
7 6 5

D. Fast Sum

Time limit: 1 second
Memory limit: 256 MB

Find the sum from 1 to N efficiently.

Input:
- First line: T (1 ≤ T ≤ 10^4)
- Next T lines: Each line contains integer N (1 ≤ N ≤ 10^6)

Output:
- Sum from 1 to N

Example:
Input:
3
2
5
10

Output:
3
15
55

E. Bubble Sort?

Time limit: 1 second
Memory limit: 256 MB

Modify the Bubble Sort algorithm to run in θ(n) time in the best case.

Input:
- First line: Integer N (1 ≤ N ≤ 10^5)
- Second line: N integers

Output:
- Sorted array

Example:

Input:
5
3 2 1 4 5

Output:
1 2 3 4 5

F. Sorting Again??

Time limit: 1 second
Memory limit: 256 MB

Sort students based on marks. If marks are equal, use Student ID.

Input:
- First line: N (1 ≤ N ≤ 1000)
- Second line: N student IDs
- Third line: N corresponding marks

Output:
- First line: Minimum swaps
- Next N lines: Sorted ID and Marks

Example:
Input:
4
7 2 5 3
80 60 80 50

Output:
Minimum swaps: 2
ID: 5 Mark: 80
ID: 7 Mark: 80
ID: 2 Mark: 60
ID: 3 Mark: 50

G. Trains?

Time limit: 3 seconds
Memory limit: 256 MB

Sort train schedules by name, then latest time, then original order.

Input:
- First line: N (1 ≤ N ≤ 100)
- Next N lines: Train description

Output:
- Sorted train descriptions

Example:
Input:
3
ABCD will departure for Mymensingh at 00:30
ABC will departure for Dhaka at 17:30
ABCD will departure for Chittagong at 01:00

Output:
ABC will departure for Dhaka at 17:30
ABCD will departure for Chittagong at 01:00
ABCD will departure for Mymensingh at 00:30

LAB 02 TASK

A. Two Sum Trouble

Time Limit: 1 second
Memory Limit: 256 MB

Bob has a sorted list of **N** integers in non-decreasing order. Given a target sum **S**, determine if there are two distinct elements whose sum equals **S**.

Input

- First line: Two integers N ($1 \le N \le 10^6$), S ($1 \le S \le 10^9$)
- Second line: N integers ($1 \le a_i \le 10^9$) in non-decreasing order

Output

- Two distinct 1-based indices i and j (i < j) such that $a_i + a$□ = S
- If no such pair exists, print −1

Examples

makefile
CopyEdit
```
Input:
4 10
1 3 5 7

Output:
2 4
```

makefile
CopyEdit
```
Input:
6 18
1 5 8 9 9 10

Output:
3 6
```

makefile
CopyEdit
```
Input:
4 7
2 4 6 8

Output:
-1
```

B. A Beautiful Sorted List

Time Limit: 1 second
Memory Limit: 1024 MB

Merge two non-decreasing sorted lists into one in O(N+M) time.

Input

- Line 1: N (length of Alice's list)
- Line 2: Alice’s N integers
- Line 3: M (length of Bob's list)
- Line 4: Bob’s M integers

Output

- A merged sorted list of length N+M

Example
makefile
CopyEdit

```
Input:
4
1 3 5 7
4
2 2 4 8

Output:
1 2 2 3 4 5 7 8
```

C. Longest Subarray Sum

Time Limit: 1 second
Memory Limit: 256 MB

Find the length of the longest contiguous subarray with a sum ≤ K.

Input

- Line 1: Two integers N and K
- Line 2: N integers (array elements)

Output

- Single integer: Length of longest subarray

Example
makefile
CopyEdit

```
Input:
5 4
4 1 2 1 5

Output:
3
```

D. Can You Iterate the Binary String?

Time Limit: 0.5 seconds
Memory Limit: 256 MB

For each binary string, find the index of the **first occurrence** of 1.

Input

- First line: T ($1 \le T \le 10^4$)
- Next T lines: Binary strings S ($1 \le |S| \le 4000$)

Output

- 1-based index of first 1, or −1 if not found

Example
makefile
CopyEdit

```
Input:
3
00000
00001
```

```
11111

Output:
-1
5
1
```

E. Count the Numbers

Time Limit: 1 second
Memory Limit: 256 MB

Count how many elements fall in a given range [x, y].

Input

- Line 1: n (array size), q (number of queries)
- Line 2: n sorted integers
- Next q lines: Queries of the form x y

Output

- For each query, print count of elements in range [x, y]

Example

makefile
CopyEdit

```
Input:
5 3
10 20 20 45 79
20 50
5 45
1 100

Output:
3
4
```

5

LAB 03 TASK

A. Count the Inversion

Use **Merge Sort** to count inversions in a permutation of numbers from 1 to N.

Input

- Line 1: N ($1 \le N \le 10^5$)
- Line 2: N integers (a permutation of 1 to N)

Output

- Line 1: Total number of inversions
- Line 2: Sorted array

Example
makefile
CopyEdit
```
Input:
5
1 2 5 4 3

Output:
3
1 2 3 4 5
```

B. Pair Maximization

Use Divide and Conquer to find max value of $(A[i] + A[j])^2$ for $i < j$.

Input

- Line 1: N
- Line 2: N integers A

Output

- Single integer: Max $(A[i]+A[j])^2$

Example
makefile
CopyEdit
```
Input:
5
4 3 1 5 6

Output:
121
```

C. Fast Power Drift

Compute abmod 107a^b \mod 107abmod107

Input

- Two integers: a ($1 \le a \le 10^4$), b ($1 \le b \le 10^{12}$)

Output

- Result of abmod 107a^b \mod 107abmod107

Example
makefile
CopyEdit
```
Input:
100 3

Output:
85
```

D. Fast Series Drift

Compute (a1+a2+...+an)mod m(a^1 + a^2 + ... + a^n) \mod m(a1+a2+...+an)modm for T test cases.

Input

- T (number of test cases)
- Each of T lines: a n m

Output

- T lines with results

E. Ordering Binary Tree

Reorder array so that BST built from insertions has minimum height.

Input

- N
- N sorted integers

Output

- A reordered list that produces a balanced BST

F. 220 Trees

Given in-order and pre-order traversal, print post-order.

Input

- N
- Line 2: in-order traversal
- Line 3: pre-order traversal

Output

- Post-order traversal

LAB04 TASKS

A. Adjacency Matrix Representation

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given a directed weighted graph with N nodes and M edges. The nodes are numbered from 1 to N. Each edge represents a direct connection between two nodes. There is no self loop or multi edge.

Input

The first line contains two integers N and M ($1 \le N \le 100$, $0 \le M \le N(N-1)/2$) – the number of vertices and the total number of edges.

The next M lines will contain three integers ui vi wi ($1 \le ui, vi \le N$, $1 \le wi \le 1000$) – denoting there is an edge from node ui to vi with cost wi.

Output

The output consists of an N×N adjacency matrix representing the directed weighted graph. Each row corresponds to a node, and each column represents its directed edges to other nodes. The value at position (i, j) denotes the weight of the edge from node i to node j. If there is no edge, the value is 0.

Examples

Input

```
6 7
1 5 6
6 3 5
1 3 9
3 4 7
4 6 1
5 6 8
6 1 6
```

Output

```
0 0 9 0 6 0
0 0 0 0 0 0
0 0 0 7 0 0
0 0 0 0 0 1
0 0 0 0 0 8
6 0 5 0 0 0
```

Input

```
4 3
```

```
1 3 8

3 2 5

1 4 2
```

Output

```
0 0 8 2

0 0 0 0

0 5 0 0

0 0 0 0
```

B. Adjacency List Representation

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given a directed weighted graph with N nodes and M edges. The nodes are numbered from 1 to N. Each edge represents a direct connection between two nodes. There is no self loop or multi edge.

Input

The first line contains two integers N and M ($1 \le N \le 100$, $0 \le M \le N(N-1)/2$) – the number of vertices and the total number of edges.

The second line contains M integers u1, u2, ..., um ($1 \le ui \le N$) – the i-th integer represents the node that is one endpoint of the i-th edge.

The third line contains M integers v1, v2, ..., vm ($1 \le vi \le N$) – the i-th integer represents the other endpoint of the i-th edge.

The fourth line contains M integers w1, w2, ..., wm (1 ≤ wi ≤ 1000) — the i-th integer represents the weight of the i-th edge.

Output

Output the adjacency list representation as shown in the sample output.

Examples

Input

```
4 5
4 1 4 3 3
3 2 2 2 1
4 4 10 8 5
```

Output

```
1: (2,4)
2:
3: (2,8) (1,5)
4: (3,4) (2,10)
```

Input

```
4 4
3 3 2 4
```

2 1 1 3

9 5 8 10

Output

1:

2: (1,8)

3: (2,9) (1,5)

4: (3,10)

C. Graph Metamorphosis

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given a directed unweighted graph with N nodes in an adjacency list format. The nodes are numbered from 0 to N-1. Your task is to convert it into an adjacency matrix representation.

Input

The first line contains a single integer N (1 ≤ N ≤ 100) — the number of vertices.

The next N lines describe the adjacency list:

The i-th line starts with an integer k, indicating the number of nodes adjacent to node i.

The next k space-separated integers represent the nodes adjacent to node i.

Output

Print an N×N adjacency matrix, where the cell at row i and column j is:

- 1 if there is an edge between nodes i and j
- 0 otherwise

Examples

Input

```
5
2 1 2
1 0
1 0
1 4
1 3
```

Output

```
0 1 1 0 0
1 0 0 0 0
1 0 0 0 0
0 0 0 0 1
```

```
0 0 0 1 0
```

Input

```
CopyEdit
5
0
2 2 3
3 1 3 4
2 1 2
1 2
```

Outpu

```
0 0 0 0 0
0 0 1 1 0
0 1 0 1 1
0 1 1 0 0
0 0 1 0 0
```

D. The Seven Bridges of Königsberg

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given an undirected unweighted connected graph with N nodes and M edges. There can be self loops or multiple edges. Your task is to determine whether an Eulerian Path exists.

Input

The first line contains two integers N and M ($1 \le N \le 2 \times 10^5$, $0 \le M \le 3 \times 10^5$)

The second line contains M integers u1, u2, ..., um

The third line contains M integers v1, v2, ..., vm

The i-th edge of this graph is between ui and vi.

Output

Print "YES" if an Eulerian Path exists, otherwise print "NO".

Examples

Inpu

```
5 10

5 5 5 2 2 2 3 3 4 2

2 3 1 3 4 1 4 1 2 4
```

Output

```
YES
```

Input

```
5 4

1 4 3 2
```

4 3 2 5

Output

YES

Input

8 7

4 4 6 6 3 1 8

6 5 3 2 7 8 7

Output

NO

Input

7 6

3 5 7 6 4 2

5 7 6 4 2 1

Output

YES

E. Edge Queries

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given a directed unweighted graph with N nodes and M edges.

Input

First line: Two integers N and M ($1 \le N \le 2 \times 10^5$, $0 \le M \le 3 \times 10^5$)

Second line: M integers u1, ..., um

Third line: M integers v1, ..., vm

Each edge is from node ui to vi.

Output

Print a single line with N space-separated integers, where the i-th integer is the difference of indegree and outdegree of node i.

Examples

Input

```
5 10

2 5 4 3 2 4 3 4 1 3

5 1 5 5 1 2 2 1 3 4
```

Output

```
2 0 -2 -2 2
```

Input

```
5 4
```

```
5 3 3 2

1 1 2 4
```

Output

```
2 0 -2 1 -1
```

Input

```
8 7

7 7 7 2 1 4 1

2 6 3 4 2 8 5
```

Output

```
-2 1 1 0 1 1 -3 1
```

F. The King of Königsberg

Time Limit: 1 second
Memory Limit: 256 megabytes

You are given an N×N chessboard and the initial position (x, y) of a King.

Input

- First line: N ($1 \le N \le 2 \times 10^5$) — size of the chessboard

- Second line: x y (1 ≤ x, y ≤ N) — initial position of the King

Output

First print K — number of valid moves.
Then K lines with two integers each, sorted in ascending order.

Examples

Input

8

1 1

Output

3

1 2

2 1

2 2

Input

8

1 2

OutpuT

5

1 1

1 3

2 1

2 2

2 3

Input

8

2 2

Output

8

1 1

1 2

1 3

2 1

2 3

3 1

3 2

3 3

G. Coprime Graph

Time Limit: 2 seconds
Memory Limit: 256 megabytes

You are given an integer N. Construct an undirected graph with N nodes, where node i is connected to node j if gcd(i, j) = 1 and i ≠ j.

There are Q queries. Each query contains X and K — find the K-th smallest node connected to node X.

Input

First line: Two integers N and Q ($1 \le N \le 2000$, $1 \le Q \le 3 \times 10^5$)
Next Q lines: Each contains X and K

Output

For each query, print the K-th smallest node connected to X. If it doesn't exist, print -1.

Examples

Input

CopyEdit

5 6

1 3

3 1

4 2

5 5

3 4

5 2

Output

```
4
1
3
-1
5
2
```

Input

```
2000 3
903 24
702 563
942 50
```

Output

```
41
1829
149
```

Input

```
1 1
1 1
```

Output

```
diff

CopyEdit

-1
```

Input

```
2 1

2 1
```

Output

```
1
```

LAB 05 TASK

🧪 Task A: Can You Traverse - 1 (BFS Traversal)

Time Limit: 5 seconds
Memory Limit: 256 MB
Graph Type: Undirected, unweighted, connected (no self-loops or multi-edges)

Perform a **Breadth-First Search (BFS)** starting from node 1. Print the order in which the nodes are visited. You may print any valid BFS traversal.

📥 Input

- First line: Two integers `N M`
  (number of nodes and number of edges)
- Next `M` lines: Each line contains two integers `u v` representing an edge between node `u` and node `v`.

📤 Output

- A single line: space-separated list of nodes in the BFS traversal order, starting from node 1.

📌 Example 1

Input:
4 3
1 4
3 2
1 3

Output:
1 3 4 2

📌 Example 2

Input:
6 10
3 1
1 6
6 4
4 5
5 2
6 2
4 3
5 6
3 6
1 5

Output:
1 3 5 6 4 2

🧪 Task B: Can You Traverse - 2 (DFS Traversal)

Time Limit: 1 second
Memory Limit: 1024 MB
Graph Type: Undirected, unweighted, connected (no self-loops or multi-edges)

Perform a **Depth-First Search (DFS)** starting from node 1. Print the order in which nodes are visited. You may print any valid DFS traversal.

📥 Input

- First line: Two integers `N M`
- Second line: `M` integers `u1 u2 ... um` (first endpoint of edges)
- Third line: `M` integers `v1 v2 ... vm` (second endpoint of edges)

Each edge is between `ui` and `vi`.

📤 Output

- A single line: space-separated list of nodes in DFS traversal order starting from node 1.

📌 Example 1

Input:
```
4 3
1 3 1
4 2 3
```

Output:
```
1 3 2 4
```

📌 Example 2

Input:
```
6 8
```

1 5 3 4 6 1 6 4
5 3 4 6 2 3 3 1

Output:
1 3 4 6 2 5

🏁 Task C: Lightning McQueen (Shortest Path + Lexicographic Order)

Time Limit: 1 second
Memory Limit: 256 MB
Graph Type: Undirected, unweighted (may be disconnected)

Find the **shortest path** from source `S` to destination `D`. If multiple paths exist, return the **lexicographically smallest** one.

📥 Input

- First line: Four integers `N M S D`
- Second line: `M` integers `u1 u2 ... um`
- Third line: `M` integers `v1 v2 ... vm`

Each edge connects `ui` and `vi`.

📤 Output

- If path exists:
  Line 1: distance (number of edges)
  Line 2: space-separated path from `S` to `D`
- If no path exists:
  `-1`

📌 **Example 1**

Input:
5 10 5 3
2 1 5 3 1 4 2 4 1 4
5 5 4 5 2 2 3 1 3 3

Output:
1
5 3

📌 **Example 2**

Input:
5 4 2 5
5 1 2 4
1 3 3 2

Output:
3
2 3 1 5

📌 **Example 3**

Input:
6 6 6 5
1 2 1 5 5 3
5 1 4 2 4 2

Output:
-1

🌴 Task D: Through the Jungle (Shortest Path via Node K)

Time Limit: 1 second
Memory Limit: 256 MB
Graph Type: Directed, unweighted

Find the **shortest path** from source `S` to destination `D`, **passing through node K**. If no such path exists, print `-1`.

📥 Input

- First line: Five integers `N M S D K`
- Next `M` lines: Two integers `u v` for each edge (directed from u to v)

📤 Output

- If path exists:
  Line 1: number of edges
  Line 2: space-separated path from `S` to `D` via `K`
- If no path exists:
  `-1`

📌 Example 1

Input:
```
5 10 5 3 5
2 5
5 1
4 5
3 5
1 2
2 4
3 2
1 4
1 3
3 4
```

Output:
```
2
5 1 3
```

📌 Example 2

Input:
6 6 2 2 2
5 1
1 2
1 4
5 2
4 5
3 2

Output:
0
2

🔄 Task E: Cycle Detection (Directed Graph)

Time Limit: 1 second
Memory Limit: 256 MB
Graph Type: Directed, unweighted

Check whether the given graph contains any **cycle**. If yes, print `YES`, else print `NO`.

📥 Input

- First line: Two integers `N M`
- Next `M` lines: Two integers `u v` representing a directed edge from `u` to `v`

📤 Output

- Print `YES` if the graph has a cycle
- Print `NO` otherwise

📌 Example 1

Input:
4 7
1 3
1 2
2 4
3 1
2 3
4 3
4 1

Output:
YES

📌 Example 2

Input:
6 5
6 4
6 3
4 5
6 2
4 1

Output:
NO

💎 Task F: Diamonds Under W (2D Grid Search)

Time Limit: 1 second
Memory Limit: 256 MB

You are given a **2D grid** with rows R and columns H.
Each cell is one of:

- . → empty cell (can move)

- D → cell with diamond (can move and collect)
- # → obstacle (cannot move)

You can **start from any non-obstacle cell** and move **up/down/left/right** to collect diamonds. Find the **maximum number of diamonds** that can be collected from any valid starting cell.

📥 Input

- First line: Two integers R H
- Next R lines: each with H characters describing the grid

📤 Output

- A single integer — the **maximum number of diamonds** collectable from any valid starting point

---

📌 Example 1

Input:
5 5
.#.DD
##.#.
####D
#DDD#
#..DD

Output:
5

📌 Example 2

Input:
5 5

D####
##.D#
#..D#
###D#
..##D

Output:
3

📌 **Example 3**

Input:
1 5
D....

Output:
1

LAB -06 TASK

A. Advising

Time limit per test: 1 second
Memory limit per test: 1024 megabytes

There are N courses and M requirements ("Course A must be before Course B"). Find a valid order of courses, or -1 if none.

Input
First line: N M
Next M lines: Ai Bi

Output
Valid order or -1

Example Input
5 4
2 4
2 5
4 3
1 5

Example Output
2 4 3 1 5

Example Input
4 6
1 2
1 3
4 1
2 3
2 4
4 3

Example Output
-1

B. A Football Match

Time limit per test: 2 seconds
Memory limit per test: 1024 megabytes

Players tackle each other: Robots only tackle Humans and Humans only tackle Robots. Given tackles, find maximum possible number of Robots or Humans.

Input
First line: N M
Next M lines: ui vi

Output
Maximum number of Robots or Humans

Example Input
5 6
3 4
3 2
5 4
5 2
4 1
1 2

Example Output
3

Example Input
5 4

4 3
1 3
3 2
3 5

Example Output
4

C. The Knight of Königsberg

Time limit per test: 1 second
Memory limit per test: 256 megabytes

Given an N×N chessboard, find minimum knight moves from (x1,y1) to (x2,y2), or -1 if impossible.

Input
N
x1 y1 x2 y2

Output
Minimum moves or -1

Example Input
3
1 2 1 3

Example Output
3

Example Input
3
1 1 2 2

Example Output
-1

D. Easy Tree Queries

Time limit per test: 1 second
Memory limit per test: 1024 megabytes

Given a rooted tree with N nodes and Q queries, find subtree size of given nodes.

Input
N R
N-1 lines edges
Q
Q lines with X

Output
Subtree size per query

Example Input
4 1
3 1
1 2
4 2
3
1
4
2

Example Output
4
1
2

E. What's the Diameter?

Time limit per test: 1 second
Memory limit per test: 1024 megabytes

Given a connected tree, find two nodes with longest path and its length.

Input
N
N-1 lines edges

Output
Length
Two nodes

Example Input
5
5 1

1 4
4 2
3 2

Example Output
4
3 5

Example Input
5
1 2
5 3
3 2
2 4

Example Output
3
5 1

F. An Ancient Ordering

Time limit per test: 1 second
Memory limit per test: 256 megabytes

Given N words sorted in a custom lexicographical order, find the order of letters or print -1 if impossible.

Input
N
N lines words

Output
Letter order or -1

Example Input
3
eat
tea
ate

Example Output
eta

Example Input
6
abc
ab
p
pq
pqr
pqrs

Example Output
-1

LAB -07 TASKS

A. Shortest Path

Time limit per test: 1 second
Memory limit per test: 1024 megabytes

You are given a directed weighted graph with N nodes and M edges. Nodes are numbered from 1 to N. No self-loops or multiple edges.

There is a source S and destination D. Find shortest distance from S to D and print the path taken. If multiple shortest paths exist, print any. If no path exists, print -1.

Input
First line: N M S D
Second line: M integers u1 u2 ... uM (starting node of each edge)
Third line: M integers v1 v2 ... vM (ending node of each edge)
Fourth line: M integers w1 w2 ... wM (weights of edges)

Output
If path exists, first line shortest distance, second line nodes on path in order. Otherwise -1.

Example Input
4 3 4 2
1 3 4
2 2 3
3 4 5

Example Output
9
4 3 2

Example Input
6 5 1 5
1 4 1 6 4
2 1 6 2 6
3 3 4 3 4

Example Output
-1

Example Input
2 1 2 1
2
1
7

Example Output
7
2 1

Example Input
5 7 2 4
1 1 5 4 2 3 2
5 4 3 5 5 4 3
3 8 2 6 6 4 3

Example Output
7
2 3 4

B. Where to Meet?

Time limit per test: 2 seconds
Memory limit per test: 1024 megabytes

Alice starts from S, Bob from T in a directed weighted graph with no self-loops or multiple edges. Find node where they can meet minimizing max arrival time. They can wait at any node.

Input
First line: N M S T
Next M lines: ui vi wi

Output

Print two integers: minimum meeting time and node number. If multiple nodes, smallest node. If no meeting possible, print -1.

Example Input

```
5 5 1 5
1 2 1
2 3 1
5 3 2
1 4 2
5 4 2
```

Example Output

```
2 3
```

Example Input

```
6 5 1 5
1 2 3
4 1 3
1 6 4
6 2 3
4 6 4
```

Example Output

```
-1
```

Example Input

```
2 1 2 2
2 1 7
```

Example Output

```
0 2
```

Example Input

```
5 7 2 5
1 5 3
1 4 8
5 3 2
4 5 6
2 5 6
3 4 4
2 3 3
```

Example Output

```
3 3
```

C. Minimize the Danger

Time limit per test: 2 seconds
Memory limit per test: 1024 megabytes

Given N cities and M bi-directional roads with danger levels (weights). From city 1 to every other city, find minimum danger level, defined as max edge danger on path. If unreachable print -1. Danger for city 1 is always 0.

Input
First line: N M
Next M lines: ui vi wi

Output
N integers: minimum danger for each city from 1.

Example Input
4 3
2 1 3
2 3 5
3 4 3

Example Output
0 3 5 5

Example Input
6 5
1 2 3
1 4 5
1 6 2
2 6 3
4 6 1

Example Output
0 3 -1 2 -1 2

Example Input
2 1
2 1 5

Example Output
0 5

Example Input
5 7
1 5 3
1 4 2
5 3 1
5 4 6
5 2 5
3 4 4
3 2 8

Example Output
0 5 3 2 3

D. Beautiful Path

Time limit per test: 2 seconds
Memory limit per test: 1024 megabytes

Directed graph with N nodes and M edges (no self-loops/multi-edges). Nodes have weights. Find minimum cost path sum of node weights from S to D. If none exists, print -1.

Input
First line: N M S D
Second line: N integers w1 w2 ... wN (weights of nodes)
Next M lines: ui vi (edges)

Output
Minimum cost or -1.

Example Input
4 3 1 2
3 4 5 4
1 2
3 2
4 3

Example Output
7

Example Input
6 5 5 3
3 3 4 3 4 1
1 2

4 1
1 6
6 2
4 6

Example Output
-1

Example Input
2 1 1 1
7 6
2 1

Example Output
7

Example Input
5 7 3 5
3 8 2 6 6
1 5
1 4
5 3
4 5
2 5
3 4
2 3

Example Output
14

E. Parity Edges

Time limit per test: 1.5 seconds
Memory limit per test: 1024 megabytes

Directed weighted graph with no self-loops/multi-edges. Find shortest distance from node 1 to node N with constraint: no two consecutive edges on path have same parity (both even or both odd). Print -1 if impossible.

Input
First line: N M
Second line: M integers u1 u2 ... uM

Third line: M integers v1 v2 ... vM
Fourth line: M integers w1 w2 ... wM

Output
Minimum distance or -1.

Example Input
```
4 3
1 3 2
4 4 3
3 4 5
```

Example Output
```
3
```

Example Input
```
6 5
1 4 1 6 4
2 1 6 2 6
3 3 4 3 4
```

Example Output
```
4
```

Example Input
```
2 1
2
1
7
```

Example Output
```
-1
```

Example Input
```
5 7
1 1 4 5 2 3 2
4 5 3 4 4 5 3
3 8 2 6 6 4 3
```

Example Output
```
8
```

F. Shortest Path Revisited

Time limit per test: 3 seconds
Memory limit per test: 1024 megabytes

Bidirectional weighted graph with no self-loops/multi-edges. Find cost of the **second shortest path** from S to D (strictly greater than shortest). If none exists, print -1.

Input
First line: N M S D
Next M lines: ui vi wi

Output
Cost of second shortest path or -1.

Example Input
4 3 2 3
2 1 3
2 3 5
3 4 3

Example Output
11

Example Input
6 5 3 4
1 2 3
1 4 5
1 6 2
2 6 3
4 6 1

Example Output
-1

Example Input
2 1 2 2
2 1 5

Example Output
10

Example Input
5 7 2 5
1 5 3
1 4 2
5 3 1
5 4 6

5 2 5
3 4 4
3 2 8

Example Output
7

LAB 08 TASK

A. Friendship

Time Limit: 1 second
Memory Limit: 1024 MB
Study Material: http://www.shafaetsplanet.com/?p=763

There is a group of **N** people living in a small village. Each person has a unique ID labeled from **1 to N**. Initially, everyone has no friends.

You're given information about **K** friendships. After each friendship, you need to print the size of the friend circle that the two people belong to (after they become friends).

Input

- First line: Two integers **N**, **K** ($1 \le N, K \le 3 \times 10^5$)
- Next **K** lines: Two integers **ai**, **aj** ($1 \le ai, aj \le N, ai \ne aj$)

Output

- For each friendship, print a single integer representing the size of the friend circle after that friendship is formed.

Example Input

8 7
2 4
4 5
3 6

4 7
3 1
2 7
6 2

Example Output

2
3
2
4
3
4
7

Explanation

After each union, the size of the combined friend circle is printed. If the two people are already in the same group, the size remains unchanged.

B. Help the King!

Time Limit: 1 second
Memory Limit: 1024 MB

There are **N** cities and **M** bidirectional roads. Each road has a maintenance cost. The king wants to **minimize the total maintenance cost** while still making sure **all cities are connected**.

Input

- First line: Two integers **N**, **M** ($2 \le N \le 2\times10^5$, $1 \le M \le 3\times10^5$)
- Next **M** lines: Three integers **ui**, **vi**, **wi** ($1 \le ui, vi \le N$, $1 \le wi \le 10^6$)

Output

- Print a single integer: the **minimum total maintenance cost** that connects all cities.

Example Input

```
4 3
3 4 5
3 1 5
2 1 2
```

Example Output

```
12
```

Example Input

```
6 5
2 6 3
2 3 3
2 1 4
6 5 1
5 4 2
```

Example Output

```
13
```

C. Again MST

Time Limit: 1 second
Memory Limit: 256 MB

You are given a connected undirected weighted graph with **N** nodes and **M** edges. You need to find the **second-best Minimum Spanning Tree** (MST), which is a spanning tree with strictly greater cost than the MST, and as low as possible.

Note

- There are no self-loops or multiple edges.
- If no second-best MST exists, print −1.

Input

- First line: Two integers **N**, **M** ($2 \le N \le 1000$, $1 \le M \le 2000$)
- Next **M** lines: Three integers **ui**, **vi**, **wi** ($1 \le ui, vi \le N$, $1 \le wi \le 10^6$)

Output

- A single integer: the **total cost of the second-best MST**, or −1 if it doesn't exist.

Example Input

```
6 7
1 2 1
2 3 2
3 1 3
1 4 5
4 5 4
5 6 5
6 4 5
```

Example Output

```
18
```

Example Input

```
5 5
1 2 3
2 3 4
```

3 4 5
4 5 1
5 1 2

Example Output

11