Balloon color: GREEN

Problem A: Agent X and the Codeword Trigram

Time Limit: 1 second
Memory Limit: 256 MB

Problem statement

In the shadowy world of international espionage, you are the last hope for **Agent X**, a top cryptanalyst on a mission codenamed "Operation Echo Chamber." A notorious organization, The *Syndicate*, is transmitting secret plans through seemingly innocent public texts. Intelligence suggests The *Syndicate* hides its messages by embedding a key phrase—a **"codeword trigram"**—repeatedly throughout their communications. A trigram is a sequence of **three consecutive words**.

Your mission, should you choose to accept it, is to build a tool that can analyze The *Syndicate*'s texts and identify their most frequently used codeword trigram. This trigram is the key to foiling their plans! Here's what Agent X has gathered from a defected *Syndicate* member:

- Each communication is broken into several **"transmissions,"**—which are separated by a dot (`.`).
- A codeword trigram will **always exist entirely within a single transmission**. It will never span across the dot marker. For instance, in "I have the files. You have the money.", the words "files You have" do not form a valid trigram.
- The *Syndicate*'s agents are sloppy with capitalization. Your analysis must be **case-insensitive**.

If multiple trigrams appear with the same highest frequency, the *Syndicate*'s protocol states that the one which appears **first** in the entire communication is the true key. Your program must report this first one.

Input Format

A single block of text representing an intercepted communication from The *Syndicate*.

Output Format

The identified codeword trigram, delivered in **lowercase**, ready for Agent X's decryption machine. (If a trigram ends with a dot then you should remove the dot).

Constraints

- The communication only contains lower or uppercase English alphabets, whitespaces, and dots (`.`).
- The total size of the communication will not exceed 10 KB.

Example 1

Input

```
I came from the moon. He went to the other room. She went to the drawing room.
```

Output

```
went to the
```

Explanation

The trigram “went to the” is the only one repeated, making it the key. Excellent work, agent!

Example 2

Input

```
I love to dance. I like to dance I. like to play chess.
```

Output

```
i love to
```

Explanation

Here, “i love to” and “i like to” both appear once. According to the protocol, we must report the one that occurred first: “i love to”. The world is safe for another day, thanks to you.

Balloon color: RED

Problem B: Chariot Race

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

At the ancient ruins of *Leptis Magna*, two friends, Omar and Tariq, imagine a great chariot race. Omar's chariot moves at a speed of A meters per minute, while Tariq's chariot moves at a speed of B meters per minute.

They race for exactly T minutes. What is the distance between Omar's and Tariq's chariots at the end of the race?

Input Format

The input consists of three integers on a single line: A, B, and T, separated by spaces. A is the speed of Omar's chariot. B is the speed of Tariq's chariot. T is the total time of the race in minutes.

Output Format

Print a single non-negative integer representing the final distance between the two chariots in meters.

Constraints

- $1 \leq A, B \leq 100$
- $1 \leq T \leq 100$

Example

Input

```
10 12 5
```

Output

```
10
```

Balloon color: BLACK

Problem C: Belt Upgrade

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

At *Musrata International Airport*, the management is upgrading the baggage belt systems. Each belt has an old capacity a_i and a new capacity b_i. The belts are currently in use and contain exactly a_i units of baggage.

To upgrade a belt, all its baggage must be temporarily moved elsewhere before the new system can be installed. Once upgraded, the belt can immediately store up to b_i units of baggage, freeing up temporary space. Only one belt can be upgraded at a time, and baggage can be temporarily stored in any free space available (including space freed by previously upgraded belts). Your task is to determine the minimum amount of temporary space required to successfully upgrade all the belts without losing or dropping any baggage.

Input Format

- The first line contains a single integer n ($1 \leq n \leq 10^5$) — the number of belts.
- The next n lines each contain two integers a_i and b_i ($0 \leq a_i, b_i \leq 10^9$), representing the current and new capacities of the i-th belt.

Output Format

- Print a single integer: the minimum temporary space T so that, for some upgrade order, baggage outside belts never exceeds T at any time.

Example 1

Input:

```
2
37 3
13 94
```

Output:

```
13
```

Explanation

Upgrade (13, 94) first: need 13 temporary units. After upgrade, 94 capacity is free.
Then upgrade (37, 3) using the already freed space. Peak temporary space = 13.

Example 2

Input:

```
2
5 0
5 0
```

Output:

```
10
```

Explanation

First upgrade needs 5 temporary units. Since no belt space is freed up (belt capacity shrinks to 0), the second upgrade adds another 5 displaced units, raising the peak temporary need to 10.

Balloon color: PURPLE

Problem D: Nano and the Flowers of Libya

Time Limit: 5.5 seconds
Memory Limit: 256 megabytes

Problem Statement

Nano finally found his true love and wants to make her happy by buying flowers. He lives in **Tripoli (node 1)**, and the roads of Libya can be modeled as a **tree** with **n** nodes (cities/neighborhoods) and **n-1** bidirectional roads. Exactly **m** of these nodes have flower kiosks.

For each **i** ($1 \leq i \leq m$), there is a kiosk at node f_i with a happiness value a_i. nano wants to collect the **maximum total happiness**. He has a special off-road buggy that moves exactly **x edges per step** in a single direction (either **down** to a descendant or **up** to an ancestor). At the start, nano chooses a fixed integer speed $x > 1$ and cannot change it later.

Movement rule:
From a node **c**, nano may move to a node **v iff** one is an ancestor of the other **and** the number of edges between them is exactly **x**.
Example intuition:
Think of Tripoli as node 1 (root). nano can jump along parent–child chains in chunks of **x** edges. He **cannot** jump between cousins/siblings directly because neither is an ancestor of the other. He may visit multiple kiosks and **does not need to return to Tripoli**.

Your task is to determine:

1. the **maximum possible total happiness** nano can achieve, choosing the **largest possible x** among those that attain this maximum (nano prefers higher speed), and
2. the **minimum number of steps** required to achieve that maximum happiness **using that chosen x**.

Input

The first line contains **T** ($1 \leq T \leq 10$) — the number of test cases. For each test case:

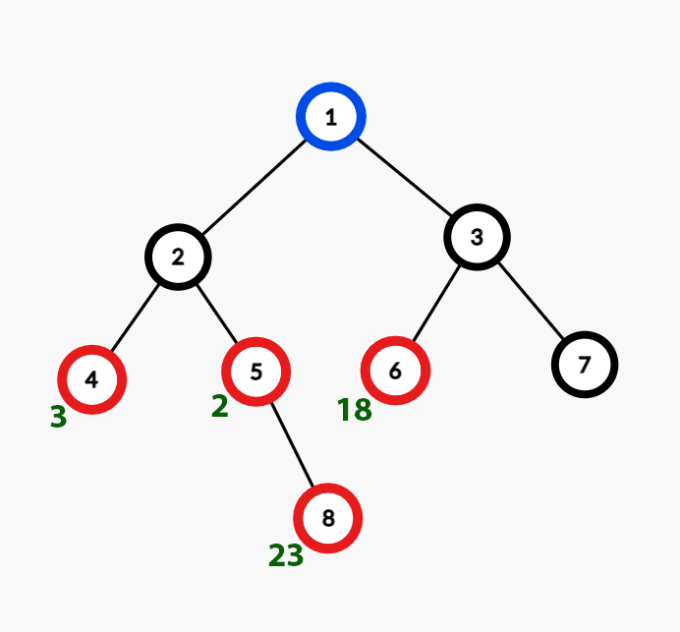
- The first line contains **n, m** ($1 \leq m < n \leq 10^5$) — the number of nodes and the number of nodes with flowers.
- The next **n-1** lines each contain **u, v** ($1 \leq u, v \leq n; u \neq v$) describing a road between nodes **u** and **v**.
- A line follows with $f_1, f_2, \ldots, f_m$ ($2 \leq f_i \leq n$).
- A final line contains $a_1, a_2, \ldots, a_m$ ($1 \leq a_i \leq 10^9$) — the happiness values at those nodes.

It is guaranteed the input forms a tree and all f_i are distinct.

Output

For each test case, print two integers:

- the **maximum total happiness** using the **largest x** that attains it, and
- the **minimum number of steps** needed to reach that happiness **with that x**.



Examples

Input

```
2
8 4
1 2
1 3
2 4
2 5
3 6
3 7
5 8
4 5 6 8
3 2 18 23
4 3
1 2
1 3
1 4
2 3 4
98 99 100
```

Output

```
23 1
0 0
```

Input

```
1
10 3
1 2
2 3
```

```
3 4
4 5
5 6
6 7
7 8
8 9
9 10
3 4 10
10 5 5
```

Output

```
10 3
```

Notes

- In the first test case, choosing $\mathbf{x = 3}$ lets nano jump directly from Tripoli (1) to a kiosk at depth 3 in **one** step and collect the best total happiness.
- In the second example, both **x=2** and **x=3** can yield total happiness **10**, but nano prefers the **larger speed x=3**; with that **x**, the minimum required steps is **3**.

Balloon color: YELLOW

Problem E: Asida Showdown

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

Farah is in the mood to cook Asida, a traditional Libyan dish, but faces a delicious dilemma. She has her grandmother's cherished recipe, and her friendly neighbor has also shared their family's version! Farah needs to figure out which Asida recipe (or recipes) she has the ingredients to make.

- **Grandma's Recipe:** Requires exactly 2 cups of flour, 2 cups of water, and 1 cup of butter.
- **Neighbor's Recipe:** A slightly different take, requiring 3 cups of flour, 1 cup of water, and 1 cup of butter.

Farah checks her kitchen pantry and finds she has F cups of flour, W cups of water, and B cups of butter. Your task is to help Farah determine which recipes she can make.

Input Format

The input consists of three non-negative integers on a single line, separated by spaces: F, W, and B.

- F: The number of flour cups Farah has.
- W: The number of water cups Farah has.
- B: The number of butter cups Farah has.

Output Format

Print one of the following four outcomes based on the ingredients Farah has:

- Either" if she has sufficient ingredients to make Grandma's recipe, and also sufficient ingredients to make the Neighbor's recipe.
- Grandma" if she only has enough ingredients for her Grandma's recipe but not the Neighbor's.
- Neighbor" if she only has enough ingredients for her Neighbor's recipe but not Grandma's.
- Neither" if she cannot make either recipe.

Constraints

- $0 \leq F, W, B \leq 20$

Example 1

Input

```
4 3 2
```

Output

```
Either
```

Example 2

Input

```
4 1 1
```

Output

```
Neighbor
```

Balloon color: WHITE

Problem F: Libya Bandito

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

Problem statement

In Libya, there is a long coastal road with houses stretching infinitely in both directions, located at $\ldots, -2, -1, 0, 1, 2, \ldots$. On day 0, a mysterious event started at house 0, affecting the residents there. Each following day, the effect spreads to exactly one neighboring house next to any already affected house. It is guaranteed that each day the affected houses form a continuous segment.

Let the segment starting at the l-th house and ending at the r-th house be denoted as $[l, r]$. You know that after n days, the segment $[l, r]$ became affected. Find any segment $[l', r']$ that could have been affected on the m-th day ($m \leq n$).

Input

The first line contains an integer t ($1 \leq t \leq 1000$) — the number of independent test cases.

Each test case consists of a single line containing four integers n, m, l, and r ($1 \leq m \leq n \leq 10^9$, $-n \leq l \leq 0 \leq r \leq n$, $r - l = n$).

Output

For each test case, output two integers l' and r' on a new line. If multiple solutions exist, print any.

Example

Input:

```
4
4 2 -2 2
4 1 0 4
3 3 -1 2
9 8 -6 3
```

Output:

```
-1 1
0 1
-1 2
-5 3
```

Note

In the first test case, it is possible that on day 1, 2, and 3 the interval of affected houses is $[-1, 0]$, $[-1, 1]$, $[-2, 1]$. Therefore, $[-1, 1]$ is a valid output.

Balloon color: LADYBUG

Problem G: Spiderweb Trees

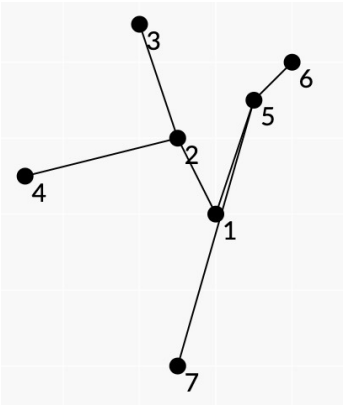
Time Limit: 1 second
Memory Limit: 256 megabytes

Problem Statement

Let's call a graph with n vertices, each of which has a point $A_i = (x_i, y_i)$ with integer coordinates, a **planar tree** if for any two non-incident edges (s_1, f_1) and (s_2, f_2), the segments $A_{s_1}A_{f_1}$ and $A_{s_2}A_{f_2}$ do not intersect. Imagine a planar tree with n vertices. Consider the convex hull of points $A_1, A_2, \ldots, A_n$. Let's call this tree a **spiderweb tree** if the following statements are true for all $1 \le i \le n$:

- All leaves (vertices with degree ≤ 1) of the tree lie on the boundary of the convex hull.
- All vertices on the boundary of the convex hull are leaves.

An example of a spiderweb tree:



The points A_3, A_6, A_7, A_4 lie on the convex hull and the leaf vertices of the tree are 3, 6, 7, 4. Refer to the notes below for more examples.

Let's call a subset $S \subset \{1, 2, \ldots, n\}$ of vertices a **subtree** of the tree if for all pairs of vertices in S, there exists a path that contains only vertices from S. Note that any subtree of a planar tree is also a planar tree.

You are given a planar tree with n vertices. Let's call a partition of the set $\{1, 2, \ldots, n\}$ into non-empty disjoint subsets $A_1, A_2, \ldots, A_k$ (i.e., $A_i \cap A_j = \emptyset$ for all $1 \le i < j \le k$ and $A_1 \cup A_2 \cup \cdots \cup A_k = \{1, 2, \ldots, n\}$) **good** if for all $1 \le i \le k$, the subtree induced by A_i is a spiderweb tree. Two partitions are different if there exists some set that lies in one partition, but not the other. Find the number of good partitions. Since this number can be very large, find it modulo 998244353.

Input

- The first line contains an integer n ($1 \le n \le 100$) — the number of vertices in the tree.
- The next n lines each contain two integers x_i, y_i ($-10^9 \le x_i, y_i \le 10^9$) — the coordinates of the i-th vertex, A_i.
- The next $n - 1$ lines contain two integers s, f ($1 \le s, f \le n$) — the edges (s, f) of the tree.

It is guaranteed that all given points are different and that no three of them lie on the same line. Additionally, it is guaranteed that the given edges and coordinates of the points describe a planar tree.

Output

Print one integer — the number of good partitions of vertices of the given planar tree, modulo 998244353.

Examples

Input

```
4
0 0
0 1
-1 -1
1 -1
1 2
1 3
1 4
```

Output

```
5
```

Input

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

Output

```
8
```

Input

```
6
4 -5
0 1
-2 8
3 -10
0 -1
-4 -5
2 5
3 2
1 2
4 6
4 2
```

Output

```
13
```

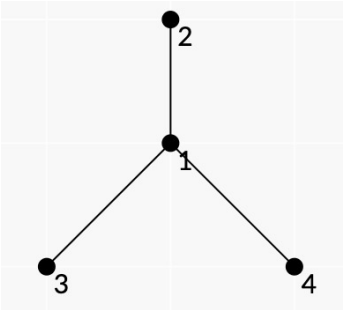
Input

```
8
0 0
-1 2
-2 5
-5 1
1 3
0 3
2 4
-1 -4
1 2
3 2
5 6
4 2
1 5
5 7
5 8
```

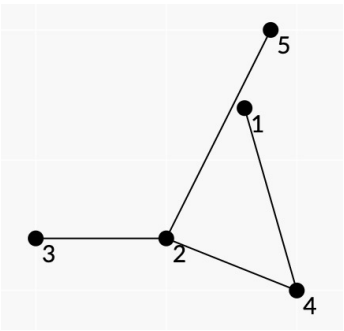
Output

```
36
```

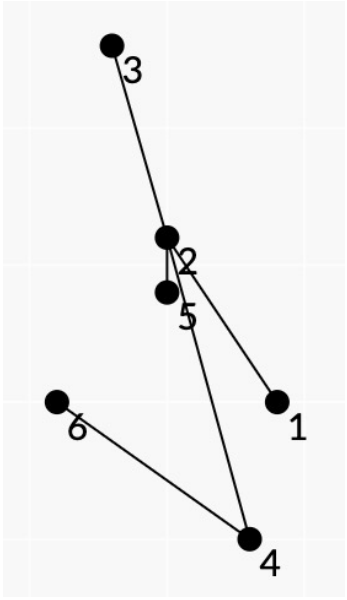
Note

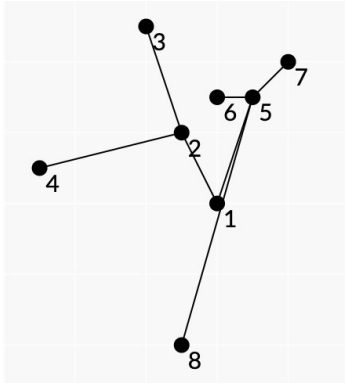


In the first test case, all good partitions are: $\{1\}, \{2\}, \{3\}, \{4\}$, $\{1,2\}, \{1,3\}, \{1,4\}$, $\{2,3\}, \{2,4\}, \{3,4\}$, $\{1,2,3,4\}$. (Note: The partition $\{1,2,3,4\}$ is a single partition, the others are not listed correctly in the original text. The correct partitions are the one-element sets and the full set, if it is a spiderweb tree.) The list of partitions in the original note seems to be garbled. The partitions are: $\{1,2,3,4\}$, $\{1,2\}, \{3\}, \{4\}$, $\{1,3\}, \{2\}, \{4\}$, $\{1,4\}, \{2\}, \{3\}$, and the partition of all singletons $\{1\}, \{2\}, \{3\}, \{4\}$. A quick check confirms these are 5 partitions. The partition $\{1,2,3\}, \{4\}$ isn't good, because the subtree $\{1,2,3\}$ isn't a spiderweb tree, since the non-leaf vertex 1 lies on the convex hull. The partition $\{2,3,4\}, \{1\}$ isn't good, because the subset $\{2,3,4\}$ isn't a subtree.



In the second test case, the given tree isn't a spiderweb tree, because the leaf vertex 1 doesn't lie on the convex hull. However, the subtree $\{2,3,4,5\}$ is a spiderweb tree.





In the fourth test case, the partition $\{1, 2, 3, 4\}, \{5, 6, 7, 8\}$ is good because all subsets are spiderweb subtrees.

Balloon color: PINK

Problem H: Maximum Water Flow

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

A private water delivery company in Libya is working to improve its daily operations. The company delivers water to different villages using tankers, trucks, and pipelines. As the company's programmer, you are tasked with solving different challenges to make the service more efficient.

The company is building a pipeline network to transport water from the main reservoir (node 1) to a large rural village (node n). Each pipeline is directional and has a maximum capacity (in cubic meters per hour). The company may send water through multiple routes at once. Your task is to calculate the maximum flow that can be sent from the reservoir to the village.

Input Format

- First line: two integers n, m — the number of nodes and the number of pipelines.
- Next m lines: three integers u, v, c — a pipeline from node u to node v with capacity c.
- The Reservoir is node 1, and the village is node n.

Output Format

- One integer — the maximum water flow from 1 to n.

Constraints

- $2 \leq n \leq 500$
- $1 \leq m \leq 5000$
- $1 \leq c \leq 1,000,000,000$

Example Run

Input:

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

Output:

```
15
```

Explanation

The maximum flow of 15 can be achieved by sending 10 units of water along the path $1 \to 2 \to 4$ and 5 units along the path $1 \to 3 \to 4$.

Balloon color: ORANGE

Problem I: Tanker Allocation Optimization

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

A private water delivery company in Libya is working to improve its daily operations. The company delivers water to different villages using tankers, trucks, and pipelines. As the company's programmer, you are tasked with solving different challenges to make the service more efficient.

The company owns water tankers, each with a specific capacity (in liters). Every day, multiple villages request water, each asking for a certain amount. Your task is to determine if all requests can be satisfied with the available tankers.

- Each tanker can serve only one village.
- A tanker must have at least the capacity requested by the village it serves.
- If all requests can be met, print "YES", otherwise print "NO".

Input Format

- First line: two integers n, m — number of tankers, number of villages ($1 \leq m, n \leq 2 \cdot 10^5$).
- Second line: n integers, c_i ($1 \leq c_i \leq 10^9$), representing the capacity of each tanker.
- Third line: m integers, r_j ($1 \leq r_j \leq 10^9$), representing the water request from each village.

Output Format

- Print "YES" if all requests can be satisfied.
- Otherwise, print "NO".

Example 1

Input:

```
3 3
500 300 200
200 300 500
```

Output:

```
YES
```

Example 2

Input:

```
3 4
600 400 300
200 200 400 500
```

Output:

```
NO
```

Balloon color: SKY BLUE

Problem J: Banknote Dilemma

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

The Central Bank of Libya has announced that certain Banknotes will expire at the end of the month. You want to use as many of these expiring notes as possible when making a payment, but you can also combine them with valid denominations.

Note: A "denomination" is the face value of a banknote. For example, the Libyan Dinar has denominations such as 1, 5, 10, 20, and 50 dinars.

You are given N denominations that will expire, and M denominations that remain valid. You are asked to make an exact payment of value P. It is guaranteed that P can always be formed using the two sets of denominations. Your goal is to maximize the number of expiring notes used in the payment.

Input Format

- First line: N, M, P
- Second line: N integers (expiring denominations)
- Third line: M integers (valid denominations)

Output Format

A single integer: maximum number of expiring notes used to pay exactly P.

Constraints

- $1 \leq N, M \leq 50$
- $1 \leq P \leq 10^4$
- Denominations are positive integers

Example

Input:

```
2 2 11
2 5
3 7
```

Output:

```
4
```

Explanation:

- Option 1: $2 + 2 + 7 = 11$ uses 2 expiring notes.
- Option 2: $2 + 2 + 2 + 5 = 11$ uses 4 expiring notes.

The maximum is 4.

Balloon color: BLUE

Problem K: Code for Solidarity

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

Watching the recent events in Gaza, a young programmer from Libya feels a powerful urge to express solidarity. He reflects on a timeless principle that, in the face of hardship, has gained a new and urgent meaning.

As he sits down for the national programming contest, he decides his first submission will be more than just code, it will be a message of unity. Your task is to write a program that broadcasts this simple but profound message.

Input Format

There is no input for this problem.

Output Format

Print the exact phrase `The Islamic World is One` on a single line.

Example

Input:

```
(There is no input)
```

Output:

```
The Islamic World is One
```

Balloon color: SILVER

Problem L: Ahmed's Journey at University of Tripoli

Time Limit: 2 seconds
Memory Limit: 512 megabytes

Problem Statement

Ahmed is an enthusiastic new student at the University of Tripoli. On his first day, he needs to get from the Main Gate to the Faculty of Information Technology for his first lecture. The campus is large and can be thought of as a network of locations (faculties, admin buildings, cafes, etc.) connected by pedestrian pathways.

You are given a map of the campus as an undirected graph. Each location is a "node," and each pathway between two locations is an "edge" with a "weight" corresponding to the time in minutes it takes to walk it. Your task is to help Ahmed find the fastest possible route from the Main Gate to the Faculty of IT by calculating the minimum total travel time.

Input Format

1. The first line contains two integers, N (the number of locations) and M (the number of pathways). Locations are numbered 1 to N.
2. The next M lines each contain three integers u, v, and w, indicating that there is a two-way pathway between location u and v which takes w minutes to walk.
3. The final line contains two integers, S (the location number of the Main Gate) and D (the location number of the Faculty of IT).

Output Format

Print a single integer representing the minimum number of minutes Ahmed needs to get from location S to D. If it's impossible to reach the destination, print -1.

Constraints

- $2 \leq N \leq 100,000$
- $1 \leq M \leq 200,000$
- $S \neq D$
- $1 \leq w \leq 1,000$

Example

Input

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

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

Output

```
9
```

Balloon color: VIOLET

Problem M: Common Ancestors

Time Limit: 2 seconds
Memory Limit: 256 megabytes

Problem Statement

You are part of a genetics research team conducting a study on the common ancestry of Libyans from different cities. The goal is to identify how much genetic material the populations of these cities share. DNA is represented as a string made up of the letters A, C, G, T. However, DNA has two special characteristics:

1. DNA is circular — meaning that the sequence has no true start or end; you can cut it at any point and it remains valid.
2. DNA can be read in reverse — the reverse orientation still represents the same genetic strand.

Your team collects two DNA samples: A from one city, B from another. You want to determine the longest genetic fragment (substring) that the two samples share, considering both rotation and reversal. This will help estimate the degree of shared ancestry. Given two DNA strings A and B, where both A and B can be rotated and/or reversed, find the length of the longest common substring between them.

Input Format

- First line: string A
- Second line: string B

Output Format

A single integer: the length of the longest common substring.

Constraints

- $1 \leq |A|, |B| \leq 500$
- Strings contain only A, C, G, T

Example

Input:

```
ACGT
GTAC
```

Output:

```
4
```

Explanation: Rotating GTAC by 2 positions gives ACGT, which perfectly matches the first strand. Thus, the longest shared substring length is 4, suggesting strong common ancestry.

Balloon color: MINT

Problem N: Benghazi Cafe Order

Time Limit: 1 second
Memory Limit: 256 MB

Problem Statement

Ahmed is ordering coffee and dates at a cafe in Benghazi's Old City. A coffee costs 5 LYD and a plate of dates costs 8 LYD. Given how many coffees and plates of dates he buys, calculate his total bill.

Input Format

- Two integers, C (coffees) and D (plates of dates).

Output Format

- One integer, T (total bill).

Constraints

- $0 \leq C, D \leq 10$

Example

Input:

```
1 2
```

Output:

```
21
```