Programming Problem Set

Problem 1: Felicity Stage Setup

You are part of the Stage Management Team for Felicity. To set up the main performance stage, you need to arrange three metal beams in a triangular shape. Based on the lengths of these beams, the stage can have different types of layouts:

- **Equilateral Layout** – All three beams are of the same length.
- **Isosceles Layout** – Exactly two beams are of the same length.
- **Scalene Layout** – All three beams are of different lengths.

Your job is to write a program that takes the lengths of the three beams and tells which layout type will be formed.

Input Format

A single line containing three space-separated integers — the lengths of the beams.

Output Format

Print one of the following on a single line:

`Equilateral` `Isosceles` `Scalene`

Examples

Input

```
5 5 5
```

Output

```
Equilateral
```

Input

```
6 8 6
```

Output

```
Isosceles
```

Input

```
4 5 6
```

Output

```
Scalene
```

Problem 2: Arjun and Coins

Arjun loves collecting coins. Every day, he either adds exactly 1 coin to his collection or spends exactly 1 coin from it. Initially, he has no coins.

After n days, can Arjun have exactly m coins in his collection?

Input Format

Each test case contains two integers n and m ($1 \leq n, m \leq 100$).

Output Format

For each test case, print `Yes` if it is possible for Arjun to have exactly m coins after n days, otherwise print `No`.

Examples

Input

```
3 3
```

Output

```
Yes
```

Input

```
2 4
```

Output

```
No
```

Input

```
5 3
```

Output

```
Yes
```

Problem 3: Distinctness

There once lived a wise ruler named Lord Sudheera, who was fascinated by the concept of distinctness. Every day, he would ponder over how many unique things existed in his kingdom. One day, while roaming his lands he came across a loyal countrymen, Vansh, and gave him a simple task:

"Vansh, I will give you a list of numbers. Since I am the LORD, each new number I give you will never be smaller than the one before - it may be greater, but never less."

Vansh, eager to please his lord, accepted the challenge. Can you help Vansh answer Lord Sudheera's queries?

Task: Given several test cases, each with a list of numbers in ascending order, determine the number of distinct elements in each list.

Input Format

- The first line contains a single integer t.
- For each test case: one line with n, then n sorted integers.

Output Format

For each test case, print a single integer — the number of distinct elements.

Examples

Input

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

Output

```
3
1
```

EXPLANATION:

In the first test case, the distinct numbers are 1, 2, and 3, so the output is 3. In the second test case, the only distinct number is 4, so the output is 1.

Problem 4: The OG Acad Office

We love acad office please don't complain to them. We love our salaries :)

Hello UG1, Welcome to the largest college in the world. This Year Acad Office was given just one task to split the batch equally yet they messed up.

Since UG1 strength is the highest of all time (largest college for a reason) acad office was given a simple task to divide students in 3 sections equally so that we can have 3 POVs of sleeping students but acad office being lazy messed up.

So they asked CPRO TA's for help but as we are also lazy so we give the task to you.

The only information we know

The number of students in each section A,B,C. We are also given that Number of students in section A ¡ Number of Students in section B ¡ Number of students in section C. Due to constraints you can only take some students out of section C and distribute them in the other two sections only. It is possible that the selected students might go in one section (obviously except C). You have to tell us whether it is possible to distribute as per the rules by the end of the deadline.

Input Format

One line containing three integers A, B, C.

Output Format

If you can split them equally, print `YES`, otherwise print `NO`.

Examples

Input

```
3 5 10
```

Output

```
YES
```

Explanation:

In section C there are 10 students and we can take 4 students from section C and put them in section A and B. We can put 3 students in section A and the last student in section B after which each section will have 6 students.

Problem 5: Fibonacci-like Sequences

You are given three integers n , x and y . You need to print the first n terms of the sequence S which starts with S 1 = x and S 2 = y and further terms are derived from the relation S i = (S i + 1) + (S i + 2)

Input Format

The only line of input contains three space-separated integers n, x, y.

Output Format

Output one line containing n space-separated integers.

Example

Input

```
5 1 2
```

Output

```
1 2 3 5 8
```

Problem 6: The Prime Bit Anomaly

Story

Deep space probe Voyager-X has detected a mysterious, repeating signal emanating from the Kepler-186f system. Our top cryptographers believe it's not random noise but a structured sequence of integers—a potential message from an extraterrestrial intelligence. The signal is weak and prone to corruption. However, our scientists have discovered the underlying generation rule, which they've named the "Prime Bit Anomaly." The rule is peculiar, combining bitwise operations with number theory.

Your mission, should you choose to accept it, is to write a program that can replicate this sequence. By providing a starting number (the first integer decoded from the signal, or seed) and a desired length, you can generate the full sequence, allowing us to analyze the complete message.

Problem Statement

You are tasked with generating a sequence of numbers based on a specific, iterative rule. The generation process starts with a given seed number and continues until the sequence reaches a specified length.

The first number in the sequence is always the seed. Every subsequent number is found by iterating through integers n (starting from n = 2) and performing a check. This check involves calculating a cumulative bitwise XOR of all numbers currently in the sequence, then XORing that result with the candidate integer n. If the count of set bits (1s) in the binary representation of this final value is a prime number, then n is the next number in the sequence.

Generation Algorithm

The sequence is built one number at a time, starting with the seed.

To find each subsequent number, a search process begins. First, a single "summary" value is calculated by applying a bitwise XOR operation to all numbers currently in the sequence.

Next, a search for the correct number to add to the sequence starts with the integer 2. For each candidate integer, a test is performed:

The candidate is combined with the sequence's "summary" value using another bitwise XOR. The number of set bits (1s) in the binary representation of the resulting value is counted. This count is checked to see if it is a prime number (e.g., 2, 3, 5, 7...). If the bit count is prime, the candidate integer is the next number in the sequence. It is added, and the process repeats to find the next number after that. If the bit count is not prime, the candidate is discarded, and the next integer (3, then 4, and so on) is tested until a valid number is found. This continues until the sequence reaches its desired length.

Input Format

A single line containing two integers: `seed` and `length`.

Output Format

Print the generated sequence as space-separated integers.

Sample Test Cases

Sample Input 1

```
10 5
```

Sample Output 1

```
10 3 2 2 2
```

Explanation 1
Start: Sequence S = [10].
Find 2nd number: Cumulative XOR $C = 10$. Try $n = 2$: $T = 10 \oplus 2 = 8$ (binary 1000), bit_count $= 1$ (not prime). Try $n = 3$: $T = 10 \oplus 3 = 9$ (binary 1001), bit_count $= 2$ (prime!). The 2nd number is 3. Sequence $S = [10, 3]$.
Find 3rd number: Cumulative XOR $C = 10 \oplus 3 = 9$. Try $n = 2$: $T = 9 \oplus 2 = 11$ (binary 1011), bit_count $= 3$ (prime!). The 3rd number is 2. Sequence $S = [10, 3, 2]$.
Find 4th number: Cumulative XOR $C = 10 \oplus 3 \oplus 2 = 11$. Try $n = 2$: $T = 11 \oplus 2 = 9$ (binary 1001), bit_count $= 2$ (prime!). The 4th number is 2. Sequence $S = [10, 3, 2, 2]$.
Find 5th number: Cumulative XOR $C = 10 \oplus 3 \oplus 2 \oplus 2 = 9$. Try $n = 2$: $T = 9 \oplus 2 = 11$ (binary 1011), bit_count $= 3$ (prime!). The 5th number is 2. Sequence $S = [10, 3, 2, 2, 2]$.
The sequence has reached length 5. **Final Output:** 10 3 2 2 2.

Sample Input 2

```
5 4
```

Sample Output 2

```
5 2 2 2
```

Explanation 2
Start: Sequence S = [5].
Find 2nd number: Cumulative XOR $C = 5$. Try $n = 2$: $T = 5 \oplus 2 = 7$ (binary 0111), bit_count $= 3$ (prime!). The 2nd number is 2. Sequence $S = [5, 2]$.
Find 3rd number: Cumulative XOR $C = 5 \oplus 2 = 7$. Try $n = 2$: $T = 7 \oplus 2 = 5$ (binary 0101), bit_count $= 2$ (prime!). The 3rd number is 2. Sequence $S = [5, 2, 2]$.
Find 4th number: Cumulative XOR $C = 5 \oplus 2 \oplus 2 = 5$. Try $n = 2$: $T = 5 \oplus 2 = 7$ (binary 0111), bit_count $= 3$ (prime!). The 4th number is 2. Sequence $S = [5, 2, 2, 2]$.
The sequence has reached length 4. **Final Output:** 5 2 2 2.