Eligibility Checker for Score Comparison

A teacher is grading two students, Alice and Bob. After evaluating their tests, she wants to quickly check whether Alice's score is greater than or equal to Bob's score to decide if Alice qualifies for a special award. Your task is to write a program that takes the scores of Alice (A) and Bob (B) and checks whether Alice's score is greater than or equal to Bob's.

- If Alice's score ≥ Bob's score, print "Yes" (Alice qualifies).
- Otherwise, print "No" (Alice does not qualify).

Input Format:

- A single line containing two space-separated integers: A B

Output Format:

- Print "Yes" if A ≥ B, otherwise print "No".

Constraints:

- 0 ≤ A, B ≤ 100

Smart Traffic Signal Decision System

Scenario

A self-driving car must decide what action to take based on the current traffic signal color:

- If the signal is **Red** → Stop
- If the signal is **Yellow** → Slow Down
- If the signal is **Green** → Go
- For any **other input** → Invalid Signal

This will help cars make safe decisions at traffic junctions.

Input Format

A single string representing the traffic signal color.
Examples: Red, Green, Yellow

Output Format

Print one of the following actions:

Stop

Slow Down

Go

Invalid Signal

Constraints

- Input is **case insensitive**
  (red / RED / ReD must work same)
- Only one signal input at a time
- No numeric values allowed

Online Login System Verification

Scenario

A user is trying to log in to an online banking application.
You must verify whether the entered **password** matches the **correct stored password**.

- If the password matches → Print **"Login Successful"**
- Otherwise → Print **"Invalid Password"**

This ensures only authorized users can access the system.

Input Format

A single string input:
The password entered by the user

Output Format

Print one of the following messages:

Login Successful

Invalid Password

Constraints

- Only **one attempt** is checked
- Password is case-sensitive
- Stored password is: "Bank@123"

Bank Account Withdrawal Validation

Scenario

A customer wants to withdraw money from their bank account. Before allowing withdrawal:

1️⃣ **First check: Is the entered amount less than or equal to available balance?**
2️⃣ **Second check: Is the amount greater than zero?**

- **If Sufficient Balance and Amount > 0 → Allow withdrawal**
- **If Amount ≤ 0 → Print "Invalid withdrawal amount"**
- **If Insufficient Balance → Print "Insufficient balance"**

This ensures secure and valid banking transactions.

Input Format

Two inputs:

1. Available balance (integer)
2. Withdrawal amount (integer)

Output Format

Print one of the following messages:

Withdrawal Successful

Invalid withdrawal amount

Insufficient balance

Constraints

- Balance ≥ 0
- Withdrawal amount can't be negative
- Integer values only

Smart Door Lock – Access Level Checker

A smart office door uses a passcode + time check. Access is granted only if:

- The passcode is "OPEN123"
- AND the time is between 8 and 19 (24-hr format)
- If the passcode is correct but the time is outside the hours → "access denied: off hours"
- If passcode is wrong → "invalid passcode"

Input Format:

- Passcode string
- Integer hour (0–23)

Output Format:

- Access message.

Sum of Last Digits

A mathematician is working with extremely large numbers and often needs a quick estimation of certain calculations. To simplify his work, he wants to determine the sum of the last digits of two given numbers without processing the entire number.

Your task is to extract the last digit of each number N and M, compute their sum, and print the result.

Input Format

- A single line containing two space-separated integers: N and M.

Output Format

- Print a single integer — the sum of the last digits of N and M.

Constraints

- 0 ≤ N, M ≤ 10^8

Check Multiples of Two Numbers

Two friends, Rahul and Simran, are working on a math game. They are given two numbers, A and B, and need to quickly check if one number is a multiple of the other to score points. Your task is to write a program that checks:

- If A is a multiple of B or B is a multiple of A, print "Multiples".
- Otherwise, print "No Multiples".

Input Format:

- A single line containing two space-separated integers: A B

Output Format:

- Print "Multiples" if one number is a multiple of the other. Otherwise, print "No Multiples".

Constraints:

- $1 \leq A, B \leq 10^9$

Case Converter

A programmer is designing a simple text editor that automatically toggles the case of a letter for user convenience.

- If the input letter is lowercase, it should be converted to uppercase.
- If the input letter is uppercase, it should be converted to lowercase.

Hint: The ASCII difference between 'a' and 'A' is 32.

Input Format:

- A single character X, which will be either a lowercase or an uppercase letter.

Output Format:

- Print the letter after changing its case.

Constraints:

- X will always be an English alphabet letter (a-z or A-Z).

Gym Fee Calculator

Ashish decides to join a local gym. The gym charges fees based on age: if Ashish is under 18, he pays 50 units; if he is between 18 and 60 (inclusive), he pays 100 units; if he is older than 60, he pays 70 units. Ashish wants to figure out how much he needs to pay to prepare his monthly budget for fitness.

Input Format:

- Single integer: age

Output Format:

- Single integer: fee

Constraints:

- 10 <= age <= 100

Complex Electricity Billing With Conflicting Premium Rules

A metropolitan electricity board has introduced an advanced billing engine to handle different consumption patterns. The system applies slab-wise billing but further adjusts rates through surcharges, penalties, and rare loyalty discounts. Because these special rules can overlap, the board enforces a strict order of evaluation to avoid conflicts. You must compute the final payable amount using the updated rules.

Billing Rules:

- First 200 units ₹3 each
- Next 200 units ₹5 each
- Remaining ₹8 each
- If units 500 15% surcharge
- If last digit is 7 add ₹200 penalty
- If units divisible by 100 AND no surcharge applied ₹300 discount
- Penalty always applies; discount only applies if surcharge does NOT.

Input format:

- An Integer U

Output format:

- Final bill (integer)

Constraints:

- 1 U 5000

Smart Traffic Signal Penalty Engine

Modern traffic systems automatically calculate penalties using multiple conditions as soon as a vehicle crosses a monitored signal. To adapt fines according to driver behavior, the system evaluates speed, vehicle type, repeated offenses, and special override alerts. These multi-conditional rules ensure fair but strict penalty assignment. You must help the traffic authorities simulate this logic and generate the correct penalty decision.

Rules:

- If speed 120 SEVERE-PENALTY
- Else if 80 speed 120 NORMAL-PENALTY

- If vehicle is "AMBULANCE" override to NO-PENALTY
- If repeatedOffender = 1 upgrade penalty one level (NORMAL SEVERE)
- If speed 20 MINOR-WARNING unless overridden
- If speed 0 INVALID

Input Format:

- An integer N representing speed
- A String representing vehicleType
- An integer representing repeatedOffender (0/1)

Output Format:

- Print the output accordingly as NO-PENALTY / SEVERE-PENALTY / NORMAL-PENALTY / MINOR-WARNING / INVALID

Constraints:

- -50 Speed 300