

1. **Scenario:** A system checks if a user is eligible to vote based on their age.

Write logic to ask the user for their age and determine if they are eligible to vote based on whether they are 18 or older.

- I. Get the age as input from the user
- II. Check the user age is above 18 or not
- III. If above 18, "Eligible to vote"
- IV. If not, "Not eligible to vote"

2. **Scenario:** A program processes a list of numbers and needs to find the largest value.

Write logic to identify and return the largest number from a given list.

- I. Assume the first number is the largest
- II. Compare it with every other number
- III. If a bigger number is found, update the largest
- IV. Return the largest value

3. **Scenario:** A company provides employees with a 10% bonus if their salary exceeds \$50,000.

Write logic to determine the bonus amount based on the given salary.

- I. Get the employee's salary details
- II. If the employee salary is greater than 50000, then add 10% bonus
- III. Else, no bonus

4. **Scenario:** A program evaluates a number to determine if it is even or odd.

Write logic to check whether a given number is even or odd.

- I. Get a number as input from the user
- II. If the number is divided by 2 with a remainder of 0, the show as even
- III. Else, show as odd

5. **Scenario:** A text-processing tool reverses a given word or sentence for formatting purposes.

Write logic to take a word or sentence as input and produce its reversed version.

- I. Get a word as text input from the user
- II. Use the reverse syntax to the input(text[::-1])
- III. Print the reversed text

6. **Scenario:** A grading system determines whether a student has passed or failed based on their score.

Write logic to check if a student has passed a subject by scoring at least 40 marks.

- I. Get the students mark as int input from the user
- II. If the student marks is above 10, print pass
- III. Else, fail

7. **Scenario:** A retail store offers a 20% discount if a customer's total order exceeds \$100. Write logic to calculate the final amount to be paid after applying the discount.

- I. Get the customers total order value as float input from the user
- II. If the customer order value exceeds 100, then calculate the 20%
- III. Reduce the discount against the final amount
- IV. Show them the final value

8. **Scenario:** A banking system processes withdrawal requests and ensures the user has enough balance.

Write logic to check if a user has enough balance before allowing a withdrawal and update the remaining balance accordingly.

- I. Get the user withdrawal amount value
- II. Check their account balance
- III. If the withdrawal amount is less that acc balance, then allow to withdraw
- IV. Else if less than acc balance, show them "Insufficient fund"
- V. If withdrawal success, show the remaining balance by withdrawal – account = New balance

9. **Scenario:** A calendar system verifies whether a given year is a leap year based on standard leap year rules.

Write logic to determine whether a given year is a leap year.

- I. Get the year details as int input from the user
- II. Check the rule, year divides by 4 with remainder 0 / divided by 100 not equal to 0 and divisible by 400 with remainder 0
- III. If Above conditions pass, its leap year
- IV. Else, its not leap year

10. **Scenario:** A program filters out only even numbers from a given list.

Write logic to extract and return only the even numbers from a list.

- I. Get the list of number
- II. Create a even rule, in a for loop If the number divides by 2 with remainder of 0
- III. Then print/show the even numbers