

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.

**Logic-1:**

1. get user age as input from user.
2. check user age is above or equal to 18
3. If above or equal to 18 means output is 'you are eligible for vote.'
4. Below means output is 'you are not eligible for 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.

**Logic-2:**

1. get list of numbers which we need to find the largest number.
2. sort the list as ascending or descending Order.
3. If ascending ,last index value is largest one  
If descending , first index value is largest one

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.

**Logic-3:**

1. Get the salary count from user.
2. Check whether it is exceed \$50,000 or not.
3. If it exceed that amount we have to increase 10% of their salary .  
 $(\text{bonus} = (10/100) * \text{salary})$   
 $\text{total salary} = \text{salary} + \text{bonus}$

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.

**Logic-4:**

1. Get the number from user.
2. Check the number is divisible by 2 or not.
3. If yes means even, no means 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.

**Logic-5:**

1. Get the sentence or word.
2. Using `[::-1]` we can reverse the sentence and word.

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.

**Logic-6:**

1. Get the mark from the user.
2. Check whether it is equal and above 40 or not.
3. If yes means pass otherwise 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.

**Logic-7:**

1. Get the total amount from the user.
2. Check whether, total amount is exceed or equal to \$100.
3. If yes means find the discount amount  
$$[\text{discount\_amount} = (20 / 100) * \text{total\_amount}]$$
4.  $\text{total\_amount} - \text{discount\_amount} = \text{final amount}.$

- 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.

**Logic-8:**

1. **Get the withdrawal amount from the user.**
2. **Check the user has enough amount or not.**
3. **If yes means, total amount - withdrawal amount = balance amount**
4. **If no means, send the text message 'insufficient 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.

**Logic-9:**

1. **Get the input year from the user.**
2. **Check whether,**  
    **year divisible by 400, or**  
    **year not divisible by 100 and**  
    **year divisible by 4.**
3. **If satisfy those condition return leap year else 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.

**Logic-10:**

1. **Get the number from the user.**
2. **Check whether it is divisible by 2 or not.**
3. **If yes, return the number.**