

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.
  - ◆ Ask the user for their age
    - Please enter your age:
  - ◆ Check if the user is 18 or above
    - if yes print ("You are eligible to vote.")
    - if not print ("Sorry, you are not eligible to vote yet.")
  
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.
  - ◆ list of numbers.
  - ◆ Assume the first number is the largest.
  - ◆ Iterate and compare.
  - ◆ Update the largest value.
  - ◆ Return the final largest number.
    - print("The largest number is:", largest)
  
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.

  - ◆ Ask for the employee's salary
    - Enter your salary:
  - ◆ If the salary is greater than \$50,000 or above
    - if yes, the bonus is 10% of the salary.
    - If not, the bonus is \$0.
  - ◆ Display the bonus amount
    - print("Your bonus is: \$", 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.

- Ask the user to enter a number
  - ◆ Enter a number:
- Check if the number is divisible by 2
  - ◆ if yes print ("The number is even.")
  - ◆ if not print ("The number is 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.

- Ask the user for input
  - ◆ Enter a word or sentence:
- Reverse the input using slicing
  - ◆ `reversed_text = text[::-1]`
- Display the reversed result
  - ◆ `print("Reversed version:", 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.

- Ask the user to enter the student's score
  - ◆ Enter the student's score:
- Check if the score is 40 or above
  - ◆ `print("The student has passed.")`
  - ◆ if not `print("The student has failed.")`

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.

- ◆ Ask the user for the total order amount
  - Enter the total order amount:
- ◆ Check if the order is \$ 100 or above, calculate a 20% discount
- ◆ Calculate the final amount to be paid
  - Final amount = Total – Discount
- ◆ Display the result

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.

- ◆ Ask the user for their current balance and withdrawal amount
  - Enter your current balance:
  - Enter the withdrawal amount:
- ◆ If the withdrawal amount is less than or equal to the balance, process the withdrawal
- ◆ Subtract the withdrawal amount from the balance and return the updated balance
- ◆ If the withdrawal amount exceeds the balance
  - `print("Insufficient funds. Withdrawal denied.")`

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.

- ◆ Ask the user to enter a year
  - Enter a year:
- ◆ Check leap year conditions
  - If divisible by 4 and not divisible by 100, or it is divisible by 400
    - ◆ `print(year, "is a leap year.")`
  - If not print(year, "is not a 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.

- List of numbers.
- Create an empty list to store even numbers.
- Iterate and check for even numbers divisible by 2.
- If divisible, add the number to the new list.
- Display the result
  - ◆ `print("Even numbers:", even_numbers)`