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.

**Answer:** ask the user for their age, check with conditions 18 or older, if 18 or below 18 not eligble, or above 18 its eligible.

**Code:**
```
age = 18
If (age==18):
print("eligible to vote")
else:
Print("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.

**Answer:** In given list, Assume the first number in the list is the largest. Go through each number in the list. If a number is larger than the current largest, update the largest. After checking all numbers, return the largest one.

**Code:**
```
list=[24,36,83,94,300]
 for num in list:
   if num>Largest:
      Largest =num
 print(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.

**Answer:** Check if the salary is more than $50,000. If it is, calculate 10% of the salary as the bonus. If it is not, the bonus is $0. Return the bonus amount.

**Code:**
```
sal=[10000,30000,60000,55000,35000]
 bonuses=[]
  for salary in sal:
      if salary>50000:
```

```
        Bonus=salary*0.10
    else:
        Bonus=0
    bonuses.append(Bonus)
print("salaries:",sal)
print("Bonuses:",bonuses)
```

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.

**Answer:** Take a number as input or enter the number. Check if the number divided by 2 leaves a remainder of 0. If it does, the number is even. If it does not, the number is odd. Use an if-else condition to make this decision and give the result.

**Code:**
```
num=23
if(num%2==0):
    print("Even")
else:
    print("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.

**Answer:** Enter word as input. Reverse the input text using the code text[::-1] takes the text and reads it **from end to start**, effectively reversing the word or sentence and Print the result.

**Code:**
```
text=input("Enter the word:")
reversed_text=text[::-1]
print("Reversed:",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.

**Answer:** Enter the students' marks. Use a for loop and if condition to determine whether each student has passed or failed.

**Code:**
```
student1=35
student2=40
student3=70
student4=90
student5=38
students=[("student1",student1),("student2",student2),("student3",student3),("student4",student4),("student5",student5)]
for name,marks in students:
    if (marks>=40):
        print(name,":Pass")
    else:
        print(name,":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.

**Answer:** Get the total order amount from the user. If it is more than $100, calculate a 20% discount and subtract it from the total using an if condition. Print the final amount to be paid.

**Code:**
```
total=500
discount=.20
print("Total:$",total)
if total>100:
        discounts=total*discount
        Final_Amount=total-discounts
         print("Discount applied:$",discounts)
else:
         print("No Discount Applied")
print("Final_Amount:$",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.

**Answer:** Read the account balance and withdrawal amount. Loop through each account holder and check withdrawal conditions. If the withdrawal amount exceeds the balance, print "Insufficient funds." Subtract the withdrawal amount from the balance and print success message. Return the final balances and withdrawal amounts.

**Code:**
```
def acdetail():
    ac_holders = {"Revi": 500,"sharmi": 5000,"shiri": 3000}
    amount = {"Revi":5000,"sharmi":1000,"shiri":500}
    for name, balance in ac_holders.items():
        withdraw_amt = amount.get(name, 0)
        if balance < withdraw_amt:
            print(f"{name}: Insufficient funds")
        else:
            ac_holders[name] -= withdraw_amt
            print(f"{name}: Withdrawal successful. Remaining balance: {ac_holders[name]}")
    return {"final_balances": ac_holders, "withdrawals": amount}
```

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.

**Answer:** Enter the year, enter the leap year condition using if else condition.

**Code:**
```
year = int(input("Enter a year: "))
if year % 4 == 0 and (year % 100 != 0 or year % 400 == 0):
    print("Leap year")
else:
    print("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.

**Answer:** Get the list of numbers, Loop through each number and check if it is divisible by 2. If the condition is true, add the number to a separate list of even numbers. After completing the loop, print the list of even numbers as the output.

**Code:**

```
number=[3,26,38,35,29,39,60]
        even_numbers=[]
        for num in number:
                if num % 2 == 0:
                        even_numbers.append(num)

        print("Even Number:",even_numbers)
```