```
"""
Task 1: Electricity Bill Calculator
Create a function calculate_bill(units):
• If units ≤ 100 → ₹1/unit
• 101-200 → ₹2/unit
• 200 → ₹3/unit
Return the total bill amount.
"""

def calculate_bill(units):
    if units <= 100:
        return units * 1
    elif units <= 200:
        return units * 2
    else:
        return units * 3

units = int(input("Enter units consumed: "))
bill = calculate_bill(units)
print("Total Bill Amount: ₹", bill)

"""
Task 2: Password Strength Checker
Write a function check_password(password) that checks:
• Length ≥ 8
• Contains at least one digit
• Contains at least one special character
Return "Strong" or "Weak".
"""

def check_password(password):
    if (len(password) >= 8 and
        any(char.isdigit() for char in password) and
        any(char in "!@#$%^&*" for char in password)):
        return "Strong"
```

```
    else:
        return "Weak"

pwd = input("Enter password: ")
print("Password Strength:", check_password(pwd))

"""
Task 3: Reverse a Number Using Loop
•  Input a number and reverse it using a while loop.
"""

num = int(input("Enter a number: "))
reverse = 0

while num > 0:
    digit = num % 10
    reverse = reverse * 10 + digit
    num = num // 10

print("Reversed Number:", reverse)

"""
Task 4: Count Vowels in a String
•  Using a for loop, count how many vowels are present in a given string.
"""

text = input("Enter a string: ")
count = 0

for ch in text:
    if ch.lower() in "aeiou":
        count += 1

print("Number of vowels:", count)
```

```
"""
Task 5: ATM Withdrawal System
Input:
• Account balance
• Withdrawal amount
Conditions:
• Amount should be a multiple of 100
• Amount ≤ balance
Display success or error message.
"""

balance = int(input("Enter account balance: "))
amount = int(input("Enter withdrawal amount: "))

if amount % 100 != 0:
    print("Error: Amount must be multiple of 100")
elif amount > balance:
    print("Error: Insufficient balance")
else:
    print("Withdrawal successful")

"""
Task 6: Student Grade with Remarks
Based on marks:
• ≥90 → A (Excellent)
• 75-89 → B (Very Good)
• 60-74 → C (Good)
• <60 → Fail
"""

marks = int(input("Enter marks: "))

if marks >= 90:
    print("Grade A - Excellent")
elif marks >= 75:
```

```
    print("Grade B - Very Good")
elif marks >= 60:
    print("Grade C - Good")
else:
    print("Fail")

"""
Task 7: Mobile Phone Class
Create a Mobile class with:
• brand
• model
• price
Methods:
• display_details()
"""

class Mobile:
    def __init__(self, brand, model, price):
        self.brand = brand
        self.model = model
        self.price = price
    def display_details(self):
        print("Brand:", self.brand)
        print("Model:", self.model)
        print("Price:", self.price)

m1 = Mobile("Samsung", "S23", 75000)
m1.display_details()

"""
Task 8: Inheritance - Employee Salary
• Base class: Employee (name, id)
• Derived class: PermanentEmployee (basic_salary)
• Method to calculate salary
"""
```

```
class Employee:
    def __init__(self, name, emp_id):
        self.name = name
        self.emp_id = emp_id

class PermanentEmployee(Employee):
    def __init__(self, name, emp_id, basic_salary):
        super().__init__(name, emp_id)
        self.basic_salary = basic_salary

    def calculate_salary(self):
        print("Total Salary:", self.basic_salary)

emp = PermanentEmployee("Rahul", 101, 50000)
emp.calculate_salary()

"""
Task 9: Palindrome Checker (Number & String)
Use:
```

- Function
- Loop
- Conditional

```
Check if input is palindrome.
"""

def check_palindrome(value):
    original = str(value)
    reverse = ""

    for ch in original:
        reverse = ch + reverse

    if original == reverse:
        print("Palindrome")
    else:
```

```
    print("Not Palindrome")


value = input("Enter number or string: ")
check_palindrome(value)
```

Outputs: