Start coding or generate with AI.

List Assignmnet

```
# Task 1: Print alternate elements of a list
def alternate_elements():
    user_list = input("Enter a list of elements separated by space: ").split()
    print("Alternate elements are:", user_list[::2])

# Task 2: Reverse the list without using reverse()
def reverse_list():
    user_list = input("Enter a list of elements separated by space: ").split()
    print("Reversed list is:", user_list[::-1])

# Task 3: Find the largest number in a list
def largest_number():
    user_list = list(map(int, input("Enter numbers separated by space: ").split()))
    largest = user_list[0]
    for num in user_list:
        if num > largest:
            largest = num
    print("The largest number is:", largest)

# Task 4: Rotate elements of a list
def rotate_list():
    user_list = input("Enter a list of elements separated by space: ").split()
    rotated_list = user_list[1:] + user_list[:1]
    print("Rotated list is:", rotated_list)

# Task 5: Delete a given word from a string
def delete_word():
    user_string = input("Enter a string: ")
    word_to_delete = input("Enter the word to delete: ")
    updated_string = user_string.replace(word_to_delete, "").strip()
    print("Updated string is:", updated_string)

# Task 6: Convert date format
def convert_date_format():
    date_string = input("Enter a date in mm/dd/yyyy format: ")
    months = ["January", "February", "March", "April", "May", "June",
              "July", "August", "September", "October", "November", "December"]
    mm, dd, yyyy = map(int, date_string.split("/"))
    print(f"{months[mm - 1]} {dd}, {yyyy}")

# Task 7: Capitalize each word
def capitalize_words():
    user_string = input("Enter a string: ")
    capitalized = " ".join(word.capitalize() for word in user_string.split())
    print("Capitalized string is:", capitalized)

# Task 8: Sum of each row of a matrix
def sum_of_rows():
    m = int(input("Enter number of rows: "))
    n = int(input("Enter number of columns: "))
    matrix = []
    for i in range(m):
        row = list(map(int, input(f"Enter row {i + 1} elements separated by space: ").split()))
        matrix.append(row)
    for i in range(m):
        print(f"Sum of row {i + 1} = {sum(matrix[i])}")

# Task 9: Add two matrices
def add_matrices():
    n = int(input("Enter number of rows: "))
    m = int(input("Enter number of columns: "))
    print("Enter elements for Matrix A:")
    matrix_a = [list(map(int, input().split())) for _ in range(n)]
    print("Enter elements for Matrix B:")
    matrix_b = [list(map(int, input().split())) for _ in range(n)]
    result = [[matrix_a[i][j] + matrix_b[i][j] for j in range(m)] for i in range(n)]
    print("Resultant Matrix:")
    for row in result:
        print(row)

# Task 10: Multiply two matrices
def multiply_matrices():
    n = int(input("Enter number of rows for Matrix A: "))
    m = int(input("Enter number of columns for Matrix A (and rows for Matrix B): "))
    p = int(input("Enter number of columns for Matrix B: "))
    print("Enter elements for Matrix A:")
    matrix_a = [list(map(int, input().split())) for _ in range(n)]
    print("Enter elements for Matrix B:")
    matrix_b = [list(map(int, input().split())) for _ in range(m)]
    result = [[sum(matrix_a[i][k] * matrix_b[k][j] for k in range(m)) for j in range(p)] for i in range(n)]
    print("Resultant Matrix:")
    for row in result:
        print(row)

# alternate_elements()
# reverse_list()
# largest_number()
# rotate_list()
# delete_word()
# convert_date_format()
# capitalize_words()
# sum_of_rows()
# add_matrices()
# multiply_matrices()
```

Looping Structures

```
# 1. Print numbers from 1 to 10 using a for loop
for i in range(1, 11):
    print(i)

# 2. Print numbers from 20 to 1 using a while loop
n = 20
while n >= 1:
    print(n)
    n -= 1

# 3. Print even numbers from 1 to 10
for i in range(1, 11):
    if i % 2 == 0:
        print(i)

# 4. Prompt user for n and print numbers from 1 to n
n = int(input("Enter a number: "))
for i in range(1, n + 1):
    print(i)

# 5. Print all odd numbers between 1 and n
n = int(input("Enter a number: "))
for i in range(1, n + 1):
    if i % 2 != 0:
        print(i)

# 6. Print 'Happy Birthday!' five times
for _ in range(5):
    print("Happy Birthday!")

# 7. Generate the first n terms of the series by squaring natural numbers
n = int(input("Enter a number: "))
print(f"The first {n} terms of the series are:")
for i in range(1, n + 1):
    print(i ** 2, end=" ")
print()

# 8. Print the multiplication table of a number
num = int(input("Enter a number: "))
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")

# 9. Print the first 8 terms of an arithmetic progression starting with 3 and common difference 4
start = 3
diff = 4
terms = 8
for i in range(terms):
    print(start + i * diff, end=" ")
print()

# 10. Print the first 6 terms of a geometric sequence starting with 2 and ratio 3
start = 2
ratio = 3
terms = 6
for i in range(terms):
    print(start * (ratio ** i), end=" ")
print()

# 11. Sum of integers from 1 to a given number
n = int(input("Enter a positive integer: "))
sum_total = sum(range(1, n + 1))
print(f"The sum of integers from 1 to {n} is: {sum_total}")

# 12. Sum of reciprocals from 1 to N
n = int(input("Enter a positive integer: "))
reciprocal_sum = sum(1 / i for i in range(1, n + 1))
print(f"The sum of reciprocals from 1 to {n} is: {reciprocal_sum:.2f}")

# 13. Accumulate numbers entered 5 times
total = 0
for _ in range(5):
    num = int(input("Enter a number: "))
    total += num
print(f"The final running total is: {total}")

# 14. Calculate the factorial of a number
n = int(input("Enter a positive integer: "))
if n < 0:
    print("Factorial does not exist for negative numbers.")
elif n == 0:
    print("The factorial of 0 is 1.")
else:
    factorial = 1
    for i in range(1, n + 1):
        factorial *= i
    print(f"The factorial of {n} is: {factorial}")

# 15. Calculate the power of a base number to an exponent
base = int(input("Enter the base number: "))
exponent = int(input("Enter the exponent: "))
result = 1
if exponent > 0:
    for _ in range(exponent):
        result *= base
elif exponent < 0:
    for _ in range(-exponent):
        result *= base
    result = 1 / result
print(f"{base} raised to the power {exponent} is: {result}")
```

Strings Assignments

```
# Python String Challenges

# 1. Count Vowels in a String
def count_vowels(string):
    """
    Count the number of vowels in the given string.
    Vowels are a, e, i, o, u (both lowercase and uppercase)
    """
    vowels = 'aeiouAEIOU'
    vowel_count = sum(1 for char in string if char in vowels)
    print(f"Number of vowels: {vowel_count}")
    return vowel_count

# 2. Count Different Character Types in a String
def count_character_types(string):
    """
    Count uppercase letters, lowercase letters, digits, and whitespace characters
    """
    uppercase_count = sum(1 for char in string if char.isupper())
    lowercase_count = sum(1 for char in string if char.islower())
    digit_count = sum(1 for char in string if char.isdigit())
    whitespace_count = sum(1 for char in string if char.isspace())

    print(f"Uppercase letters: {uppercase_count}")
    print(f"Lowercase letters: {lowercase_count}")
    print(f"Digits: {digit_count}")
    print(f"Whitespace characters: {whitespace_count}")

# 3. Exchange First and Last Characters
def exchange_first_last(string):
    """
    Create a new string by exchanging the first and last characters
    """
    if len(string) <= 1:
        return string
    new_string = string[-1] + string[1:-1] + string[0]
    print(f"Original string: {string}")
    print(f"Modified string: {new_string}")
    return new_string

# 4. Reverse a String
def reverse_string(string):
    """
    Create a new string by reversing the original string
    Note: Not using reverse() method
    """
    reversed_str = ''
    for i in range(len(string) - 1, -1, -1):
        reversed_str += string[i]
    print(f"Original string: {string}")
    print(f"Reversed string: {reversed_str}")
    return reversed_str

# 5. Shift String One Position Left
def shift_left(string):
    """
    Create a new string by shifting characters one position to the left
    """
    if len(string) <= 1:
        return string
    new_string = string[1:] + string[0]
    print(f"Original string: {string}")
    print(f"Shifted string: {new_string}")
    return new_string

# 6. Print Initials
def print_initials(full_name):
    """
    Print initials of a full name
    Note: Not using split() method
    """
    initials = ''
    initials += full_name[0] + '.'

    for i in range(len(full_name)):
        if full_name[i] == ' ':
            initials += full_name[i+1] + '.'

    print(f"Full Name: {full_name}")
    print(f"Initials: {initials}")
    return initials

# 7. Palindrome Checker
def is_palindrome(string):
    """
    Check if a string is a palindrome
    Note: Not using reverse() method
    """
    # Compare characters from start and end moving inwards
    left = 0
    right = len(string) - 1

    while left < right:
        if string[left] != string[right]:
            return False
        left += 1
        right -= 1

    return True

# 8. Circular Shift Display
def circular_shift_display(string):
    """
    Display circular shifts of the string
    """
    for i in range(len(string)):
        shifted = string[i:] + string[:i]
        print(shifted)

# 9. Password Validation
def validate_password(password):
    """
    Validate password based on specific requirements:
    - At least 8 characters long
    - Contains at least one uppercase letter
    - Contains at least one lowercase letter
    - Contains at least one numeric digit
    """
    # Check length
    if len(password) < 8:
        print("Password must be at least 8 characters long.")
        return False

    # Check for uppercase, lowercase, and digit
    has_uppercase = False
    has_lowercase = False
    has_digit = False

    for char in password:
        if char.isupper():
            has_uppercase = True
        elif char.islower():
            has_lowercase = True
        elif char.isdigit():
            has_digit = True

    # Validate all conditions
    if not has_uppercase:
        print("Password must contain at least one uppercase letter.")
        return False

    if not has_lowercase:
        print("Password must contain at least one lowercase letter.")
        return False

    if not has_digit:
        print("Password must contain at least one numeric digit.")
        return False

    print("Password is valid!")
    return True

# Demonstration of all functions
def main():
    print("1. Vowel Counter")
    count_vowels("Hello World")

    print("\n2. Character Type Counter")
    count_character_types("Hello World 123")

    print("\n3. First and Last Character Exchange")
    exchange_first_last("HELLO")

    print("\n4. String Reversal")
    reverse_string("EXAM")

    print("\n5. Left Shift")
    shift_left("examination 2021")

    print("\n6. Initials")
    print_initials("Ajay Kumar Garg")

    print("\n7. Palindrome Check")
    print(f"'madam' is palindrome: {is_palindrome('madam')}")
    print(f"'hello' is palindrome: {is_palindrome('hello')}")

    print("\n8. Circular Shift")
    circular_shift_display("SHIFT")

    print("\n9. Password Validation")
    validate_password("StrongPass123")
    validate_password("weak")

# Uncomment the line below to run all demonstrations
# main()
```

Conditional Structures

```
# 1. Check if a number is even or odd
number = int(input("Enter a number: "))
if number % 2 == 0:
    print("The number is even.")
else:
    print("The number is odd.")
    print("The number is odd.")

# 2. Determine voting eligibility
age = int(input("Enter your age: "))
if age >= 18:
    print("You are eligible to vote.")
else:
    print("You are not eligible to vote yet.")

# 3. Find the largest of two integers
num1 = int(input("Enter the first integer: "))
num2 = int(input("Enter the second integer: "))
if num1 > num2:
    print(f"The larger number is {num1}.")
elif num2 > num1:
    print(f"The larger number is {num2}.")
else:
    print("Both numbers are equal.")

# 4. Determine if a number is positive, negative, or zero
number = int(input("Enter a number: "))
if number > 0:
    print("Positive")
elif number < 0:
    print("Negative")
else:
    print("Zero")

# 5. Determine age group
age = int(input("Enter your age: "))
if 0 <= age <= 12:
    print("Child")
elif 13 <= age <= 19:
    print("Teenager")
elif 20 <= age <= 59:
    print("Adult")
elif age >= 60:
    print("Senior Citizen")
else:
    print("Invalid age.")

# 6. Display day of the week
day = int(input("Enter a number (1-7): "))
days_of_week = ["Sunday", "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday"]
if 1 <= day <= 7:
    print(f"The day is {days_of_week[day - 1]}.")
else:
    print("Invalid input. Please enter a number between 1 and 7.")

# 7. Calculate and classify BMI
weight = float(input("Enter your weight in kilograms: "))
height = float(input("Enter your height in meters: "))
bmi = weight / (height ** 2)
if bmi < 18.5:
    category = "Underweight"
elif 18.5 <= bmi < 24.9:
    category = "Normal weight"
elif 25 <= bmi < 29.9:
    category = "Overweight"
else:
    category = "Obesity"
print(f"Your BMI is {bmi:.2f}, classified as {category}.")

# 8. Calculate average and grade
marks = [float(input(f"Enter marks for subject {i + 1}: ")) for i in range(3)]
average = sum(marks) / 3
if 90 <= average <= 100:
    grade = "A"
elif 80 <= average < 90:
    grade = "B"
elif 70 <= average < 80:
    grade = "C"
elif 60 <= average < 70:
    grade = "D"
else:
    grade = "F"
print(f"Average marks: {average:.2f}, Grade: {grade}")

# 9. Solve quadratic equation
import math
a = float(input("Enter coefficient a: "))
b = float(input("Enter coefficient b: "))
c = float(input("Enter constant c: "))
```

```
discriminant = b ** 2 - 4 * a * c
if discriminant > 0:
    root1 = (-b + math.sqrt(discriminant)) / (2 * a)
    root2 = (-b - math.sqrt(discriminant)) / (2 * a)
    print(f"Two real roots: {root1}, {root2}")
elif discriminant == 0:
    root = -b / (2 * a)
    print(f"One real root: {root}")
else:
    real_part = -b / (2 * a)
    imaginary_part = math.sqrt(-discriminant) / (2 * a)
    print(f"Two complex roots: {real_part} ± {imaginary_part}i")

# 10. Sort three numbers in ascending order
nums = [float(input(f"Enter number {i + 1}: ")) for i in range(3)]
nums.sort()
print("Sorted numbers:", nums)

# 11. Find the largest of three integers
nums = [int(input(f"Enter integer {i + 1}: ")) for i in range(3)]
print(f"The largest number is {max(nums)}.")

# 12. Determine if a character is a vowel or consonant
char = input("Enter a character: ").lower()
if char in "aeiou":
    print("The character is a vowel.")
elif char.isalpha():
    print("The character is a consonant.")
else:
    print("Invalid input.")

# 13. Determine if a year is a leap year
year = int(input("Enter a year: "))
if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):
    print(f"{year} is a Leap Year.")
else:
    print(f"{year} is NOT a Leap Year.")

# 14. Calculate telephone bills
calls = int(input("Enter the number of calls: "))
if calls <= 100:
    bill = 200
elif calls <= 150:
    bill = 200 + (calls - 100) * 0.6
elif calls <= 200:
    bill = 200 + 50 * 0.6 + (calls - 150) * 0.5
else:
    bill = 200 + 50 * 0.6 + 50 * 0.5 + (calls - 200) * 0.4
print(f"Your telephone bill is: Rs. {bill:.2f}")
```