SRM INSTITUTE OF SCIENCE &TECHNOLOGY

Kattankulathur

Chennai

Name ➢ **GAURAV GUPTA**

Subject ➢ Advanced Programming practice

Section ➢ W2

Roll No. ➢ RA2211026010284

Title ➢ Assignment Week 10

Assignment

Week 10

Q1) Implement a python program to find the first largest and second largest numbers in an Array. Note: should not use any built-in sorting functions or libraries.

CODE:

```
a=[10,20,30,40,50]
l=len(a)
fl=a[1]
f2=a[2]
for i in range(l):
    if (a[i]>fl):
        f2=fl
        fl=a[i]
print("Largest No. in array is: ",fl)
print("Second largest No. in the array is: ",f2)
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assi
gnment 9.py
Largest No. in array is:  50
Second largest No. in the array is:  40
>>>
```

Q2) Write a Python program to calculate the sum of even numbers and the sum of odd numbers in an array.

CODE:

```
a=[10,15,20,25]
l=len(a)

es=0
os=0
for i in range(l):
    print(a[i],end=' ')
    if (a[i]%2==0):
        es+=a[i]
    else:
        os+=a[i]
print("\nEven element sum is: ",es)
print("odd element sum is: ",os)
```

Output

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assi
gnment 9.py
10 15 20 25
Even element sum is:  30
odd element sum is:  40
>>>
```

Q3) Write a python program to count the Occurrences of a Specific Element in an Array.

CODE:

```
a=[10,15,20,10,25]
b=int(input("Enter the element to check "))
l=len(a)
c=0
for i in range(l):
    if (b==a[i]):
        c+=1

if (c==0):
    print("The entered element doen't exist in array")
print("Frequencey of element is: ",c)
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assi
gnment 9.py
Enter the element to check 10
Frequencey of element is:  2
>>>
```

Q4) Write a Python program that takes a sentence as input and identifies and prints all the palindromic words in the sentence. Use an array to store the palindromic words.

CODE:

```
sentence = input("Enter a sentence: ")

words = sentence.split()
palindromic_words = []

for word in words:
    if word == word[::-1]:
        palindromic_words.append(word)

if palindromic_words:
    print("Palindromic words in the sentence:")
    print(palindromic_words)
else:
    print("No palindromic words found in the sentence.")
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assignment 9.py
Enter a sentence: My name is Gaurav I am from malayalam
Palindromic words in the sentence:
['I', 'malayalam']
>>>
```

Q5) Write a Python program that takes a list of numbers and removes all duplicates from the list, preserving the original order of elements.

CODE

```
numbers = input("Enter a list of numbers separated by spaces: ").split()
numbers = [int(num) for num in numbers]
unique_list = []
for num in numbers:
    if num not in unique_list:
        unique_list.append(num)
print("List with duplicates removed:")
print(unique_list)
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assignment 9.py
Enter a list of numbers separated by spaces: 1 2 3 4 5 6 2 1 2 5  8 9
List with duplicates removed:
[1, 2, 3, 4, 5, 6, 8, 9]
>>>
```

Q6) Write a Python program that performs matrix multiplication. Ask the user to input two matrices as lists of lists (2D arrays) and then multiply them if possible. Make sure to check if the matrices are compatible for multiplication and handle errors gracefully.

CODE

```
rows1 = int(input("Enter the number of rows for the first matrix: "))
cols1 = int(input("Enter the number of columns for the first matrix: "))

rows2 = int(input("Enter the number of rows for the second matrix: "))
cols2 = int(input("Enter the number of columns for the second matrix: "))

matrix1 = []
print("Enter elements for the first matrix:")
for i in range(rows1):
    row = []
    for j in range(cols1):
        element = int(input(f"Enter element ({i+1}, {j+1}): "))
        row.append(element)
    matrix1.append(row)

matrix2 = []
print("\nEnter elements for the second matrix:")
for i in range(rows2):
    row = []
    for j in range(cols2):
        element = int(input(f"Enter element ({i+1}, {j+1}): "))
        row.append(element)
    matrix2.append(row)

if cols1 != rows2:
```

```
    print("\nMatrices are not compatible for multiplication.")
else:

    result = [[0 for _ in range(cols2)] for _ in range(rows1)]
    for i in range(rows1):
        for j in range(cols2):
            for k in range(cols1):
                result[i][j] += matrix1[i][k] * matrix2[k][j]

    print("\nResultant Matrix (after multiplication):")
    for row in result:
        print(row)
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assi
gnment 9.py
Enter the number of rows for the first matrix: 2
Enter the number of columns for the first matrix: 2
Enter the number of rows for the second matrix: 2
Enter the number of columns for the second matrix: 2
Enter elements for the first matrix:
Enter element (1, 1): 1
Enter element (1, 2): 2
Enter element (2, 1): 3
Enter element (2, 2): 4

Enter elements for the second matrix:
Enter element (1, 1): 4
Enter element (1, 2): 3
Enter element (2, 1): 2
Enter element (2, 2): 1

Resultant Matrix (after multiplication):
[8, 5]
[20, 13]
>>>
```

Q7) Write a python program to print diamond number pattern using Nested Loops.

CODE:

```
num_rows = int(input("Enter the number of rows for the diamond pattern: "))

for i in range(1, num_rows + 1, 2):
    print(" " * ((num_rows - i) // 2), end="")
    for j in range(1, i + 1):
        print(j, end="")
    print()

for i in range(num_rows - 2, 0, -2):
    print(" " * ((num_rows - i) // 2), end="")
    for j in range(1, i + 1):
        print(j, end="")
    print()
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assi
gnment 9.py
Enter the number of rows for the diamond pattern: 5
  1
 123
12345
 123
  1
>>>
```

Q8) Write a Python program that simulates a simple guessing game. Generate a random number and have the user guess it. Provide hints like "too high" or "too low" until they guess correctly.

CODE:

```
import random
secret_number = random.randint(1, 100)
num_guesses = 0

print("Welcome to the Guessing Game!")
print("Try to guess the secret number between 1 and 100.")

while True:

    user_guess = int(input("Enter your guess: "))
    num_guesses += 1

    if user_guess == secret_number:
        print(f"Congratulations! You guessed the secret number {secret_number} in {num_guesses} guesses.")
        break
```

```
    elif user_guess < secret_number:
        print("Too low! Try a higher number.")
    else:
        print("Too high! Try a lower number.")
```

Output:

```
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assignment 9.py
Welcome to the Guessing Game!
Try to guess the secret number between 1 and 100.
Enter your guess: 5
Too low! Try a higher number.
Enter your guess: 55
Too high! Try a lower number.
Enter your guess: 25
Too low! Try a higher number.
Enter your guess: 30
Too low! Try a higher number.
Enter your guess: 40
Too high! Try a lower number.
Enter your guess: 33
Too high! Try a lower number.
```

Q9) Write a Python program that checks the strength of a password entered by a user. The program should assess the password based on criteria like length, use of uppercase and lowercase letters, digits, and special characters. Use control structures and arrays to provide a detailed evaluation.

CODE:

```
def check_password_strength(password):
    # Define criteria
    length_criteria = len(password) >= 8
    uppercase_criteria = any(char.isupper() for char in password)
    lowercase_criteria = any(char.islower() for char in password)
    digit_criteria = any(char.isdigit() for char in password)
    special_char_criteria = any(char in "!@#$%^&*()-_=+[]{}|;:'\",.<>?`~" for char in password)
```

```
    criteria_messages = [
        "Minimum 8 characters",
        "At least one uppercase letter",
        "At least one lowercase letter",
        "At least one digit",
        "At least one special character: !@#$%^&*()-_=+[]{}|;:'\",.<>?`~"
    ]
    strength = "Strong"
    for i, criteria in enumerate([length_criteria, uppercase_criteria, lowercase_criteria, digit_criteria, special_char_criteria]):
        if not criteria:
            strength = "Weak"
            print(f"Password is weak. Missing: {criteria_messages[i]}")

    return strength
password = input("Enter a password: ")
result = check_password_strength(password)
if result == "Strong":
    print("Password is strong!")
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assignment 9.py
Enter a password: Gaurav@1947
Password is strong!
>>>
```

Ln: 107 Col

Q10) Write a Python program that generates the Fibonacci sequence up to a specified number of terms using a loop and stores it in an array.

CODE:

```
num_terms = int(input("Enter the number of Fibonacci terms to generate: "))

fib_sequence = [0, 1]

while len(fib_sequence) < num_terms:
    next_term = fib_sequence[-1] + fib_sequence[-2]
    fib_sequence.append(next_term)

print(f"The Fibonacci sequence with {num_terms} terms is:")
print(fib_sequence[:num_terms])
```

Output:

```
>>>
= RESTART: D:\SRM\SEMESTERS\3rd SEM\Advance Programming\python\assignment 9.py
Enter the number of Fibonacci terms to generate: 8
The Fibonacci sequence with 8 terms is:
[0, 1, 1, 2, 3, 5, 8, 13]
>>>
```