Lab 4: Python Basics

Date: 31/01/2026

Q1. Write a python program to reverse a content a file and store it in another file.

Code:

```
def reverseFile(inpf, outf):
    with open(inpf, 'r') as f:
        content = f.read()

    reversed = content[::-1]

    with open(outf, 'w') as f:
        f.write(reversed)

inpfile = "files/source.txt"
outfile = "files/reversed.txt"

reverseFile(inpfile, outfile);
print(f"Reversed content from '{inpfile}' has been written to '{outfile}'")
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 ReverseFile.py
Reversed content from 'files/source.txt' has been written to 'files/reversed.tx
t'
WPC2@selab-09:~/Documents/230905252/Lab4$ cat files/source.txt

Hello, my name is Dhruv!
This is Lab 4 of Web Programming Lab.
We are doing Python basics.
WPC2@selab-09:~/Documents/230905252/Lab4$ cat files/reversed.txt

.scisab nohtyP gniod era eW
.baL gnimmargorP beW fo 4 baL si sihT
!vurhD si eman ym ,olleH
```

2. Write a python program to implement binary search with recursion.

Code:

```
def BinarySearch(arr, target, low, high):
    if low > high:
        return -1
    mid = (low + high) // 2

    if (arr[mid] == target):
        return mid
    elif target < arr[mid]:
        return BinarySearch(arr, target, low, mid-1)
    else:
        return BinarySearch(arr, target, mid+1, high)

arr = list(map(int, input("Enter sorted elements (space separated): ").split()))
target = int(input("Enter target element: "))

result = BinarySearch(arr, target, 0, len(arr)-1)

if (result != -1):
    print(f"Element {target} found at index {result}")
else:
    print(f"Element {target} not found in array")
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 BinarySearch.py
Enter sorted elements (space separated): 1 3 5 7 9
Enter target element: 7
Element 7 found at index 3
```

3. Write a python program to sort words in alphabetical order.

Code:

```
def sort(words):
    for i in range(len(words)):
        key = words[i]
        j = i-1

        while j >= 0 and words[j].lower() > key.lower():
            words[j+1] = words[j]
            j -= 1
        words[j+1] = key
    return words

text = input("Enter words to sort: ")
sorted = sort(text.split())

print("Words in alphabetical order:")
for word in sorted:
    print(word)
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 WordSort.py
Enter words to sort: hello bye mango apple rain
Words in alphabetical order:
apple
bye
hello
mango
rain
```

4. Write a Python class to get all possible unique subsets from a set of distinct integers.

Code:

```
class SubsetGenerator:
    def __init__(self, nums):
        self.nums = nums

    def getSubsets(self):
        result = [[]]

        for num in self.nums:
            new = [subset + [num] for subset in result]
            result.extend(new)

        return result

nums = list(map(int, input("Enter distinct integers (space separated): ").split()))
generator = SubsetGenerator(nums)
subsets = generator.getSubsets()

print("All unique subsets:")
print(subsets)
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 SubsetGenerator.py
Enter distinct integers (space separated): 4 5 6
All unique subsets:
[[], [4], [5], [4, 5], [6], [4, 6], [5, 6], [4, 5, 6]]
```

5. Write a Python class to find a pair of elements (indices of the two numbers) from a given array whose sum equals a specific target number.

Code:

```
class TwoSumSolver:
    def __init__(self, arr, target):
        self.arr = arr
        self.target = target

    def findPair(self):
        seen = {}
        for i, num in enumerate(arr):
            comp = target - num;
            if comp in seen:
                return [seen[comp], i]
            seen[num] = i
        return None

arr = list(map(int, input("Enter elements (space separated): ").split()))
target = int(input("Enter target sum: "))

solver = TwoSumSolver(arr, target)
result = solver.findPair()

if result:
    print(f"Pair found at indexes {result[0]} and {result[1]}")
else:
    print("No pair found")
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 TwoSum.py
Enter elements (space separated): 10 20 10 40 50 60 70
Enter target sum: 50
Pair found at indexes 2 and 3
```

6. Write a Python class to implement pow(x, n).

Code:

```
class PowerCalculator:
    def __init__(self, x, n):
        self.x = x
        self.n = n

    def pow(self):
        x, n = self.x, self.n
        res = 1

        if n < 0:
            x = 1/x
            n = -n

        while n > 0:
            if n % 2 == 1:
                res *= x
            x *= x
            n //= 2

        return res

x = float(input("Enter base (x): "))
n = int(input("Enter exponent (n): "))

calculator = PowerCalculator(x, n)
print(f"{x}^{n} = {round(calculator.pow(),8)}")
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 PowerFunc.py
Enter base (x): 2.5
Enter exponent (n): -2
2.5^-2 = 0.16
```

7. Write a Python class which has two methods get_String and print_String. The get_String accept a string from the user and print_String print the string in upper case.

Code:

```
class StringHandler:
    def __init__(self):
        self.text = ""

    def getString(self):
        self.text = input("Enter a string: ")

    def printString(self):
        print(self.text.upper())

handler = StringHandler()
handler.getString()
handler.printString()
```

Output:

```
WPC2@selab-09:~/Documents/230905252/Lab4$ python3 StringClass.py
Enter a string: hello world
HELLO WORLD
```