NAME:AYANIKA PAUL
BATCH:D1
ROLL NO. 22

LAB 4

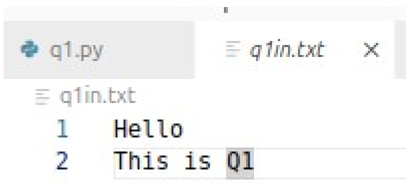
Q1)Write a python program to reverse a content a file and store it in another file.

CODE:

```
def reverse_file_content(input_file, output_file):
try:
with open(input_file, 'r') as infile:
content = infile.read()
reversed_content = content[::-1]

with open(output_file, 'w') as outfile:
outfile.write(reversed_content)

print(f"Content reversed and saved to '{output_file}' successfully.")
except FileNotFoundError:
print(f"Error: The file '{input_file}' does not exist.")
except Exception as e:
print(f"An error occurred: {e}")

input_file = 'q1in.txt'
output_file = 'output.txt'
reverse_file_content(input_file, output_file)
```

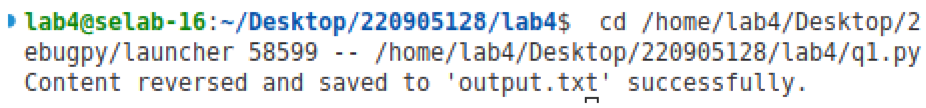
OUTPUT:

q1in.txt



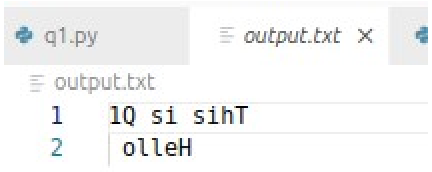
```
q1in.txt
1   Hello
2   This is Q1
```

Terminal:



```
lab4@selab-16:~/Desktop/220905128/lab4$  cd /home/lab4/Desktop/2
ebugpy/launcher 58599 -- /home/lab4/Desktop/220905128/lab4/q1.py
Content reversed and saved to 'output.txt' successfully.
```

Output.txt



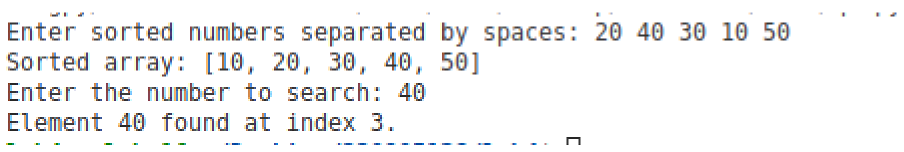
```
output.txt
1   1Q si sihT
2    olleH
```

Q2)Write a python program to implement binary search with recursion.

CODE:

```
def binary_search_recursive(arr, target, low, high):
if low > high:
return -1
mid = (low + high)
if arr[mid] == target:
return mid
elif arr[mid] > target:
return binary_search_recursive(arr, target, low, mid - 1) # left half
else:
return binary_search_recursive(arr, target, mid + 1, high) #right half

arr = list(map(int, input("Enter sorted numbers separated by spaces: ").split()))
arr.sort()
print("Sorted array:", arr)

target = int(input("Enter the number to search: "))

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

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

Output:



```
Enter sorted numbers separated by spaces: 20 40 30 10 50
Sorted array: [10, 20, 30, 40, 50]
Enter the number to search: 40
Element 40 found at index 3.
```

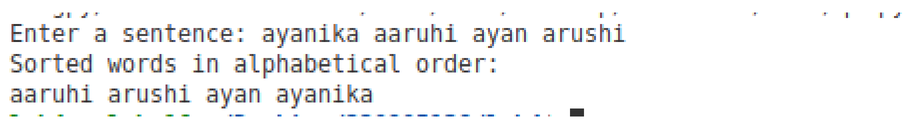
Q3)Write a python program to sort words in alphabetical order.

CODE:

```
def sort_words_alphabetically(sentence):
words = sentence.split()
```

```
words.sort()
return words
sentence = input("Enter a sentence: ")

sorted_words = sort_words_alphabetically(sentence)

print("Sorted words in alphabetical order:")
print(" ".join(sorted_words))
```

Output:



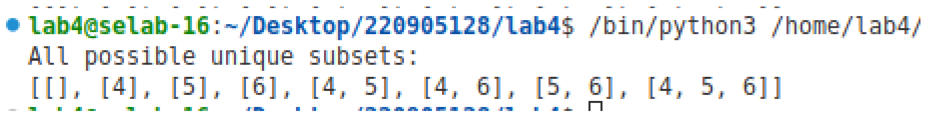
```
Enter a sentence: ayanika aaruhi ayan arushi
Sorted words in alphabetical order:
aaruhi arushi ayan ayanika
```

Q4)Write a Python class to get all possible unique subsets from a set of distinct integers Input:[4,5,6]
Output : [[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [4, 5, 6]]

CODE:

```
from itertools import combinations

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

def get_subsets(self):
subsets = []
for i in range(len(self.nums) + 1):
for subset in combinations(self.nums, i):
subsets.append(list(subset))
return subsets
nums = [4, 5, 6]

subset_generator = SubsetGenerator(nums)

result = subset_generator.get_subsets()

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

Output



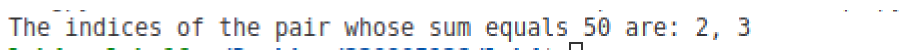
```
lab4@selab-16:~/Desktop/220905128/lab4$ /bin/python3 /home/lab4/
All possible unique subsets:
[[], [4], [5], [6], [4, 5], [4, 6], [5, 6], [4, 5, 6]]
```

Q5)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.
Input: numbers= [10,20,10,40,50,60,70], target=50

CODE:

```
class PairSumFinder:
def __init__(self, numbers):
self.numbers = numbers

def find_pair(self, target):
num_indices = {}
for index, num in enumerate(self.numbers):
complement = target - num
if complement in num_indices:
return (num_indices[complement], index)
num_indices[num] = index
return None
numbers = [10, 20, 10, 40, 50, 60, 70]
target = 50
pair_finder = PairSumFinder(numbers)
result = pair_finder.find_pair(target)
if result:
print(f"The indices of the pair whose sum equals {target} are: {result[0]}, {result[1]}")
else:
print(f"No pair found whose sum equals {target}.")
```

Output:



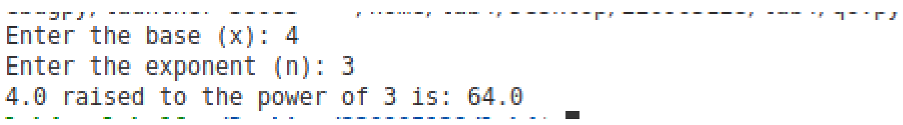
```
The indices of the pair whose sum equals 50 are: 2, 3
```

Q6)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):
if self.n < 0:
return 1 / self._pow_helper(self.x, -self.n)
else:
return self._pow_helper(self.x, self.n)

def _pow_helper(self, x, n):
if n == 0:
return 1
if n == 1:
return x
half = self._pow_helper(x, n // 2)
if n % 2 == 0:
return half * half
else:
return half * half * x
x = float(input("Enter the base (x): "))
n = int(input("Enter the exponent (n): "))

calculator = PowerCalculator(x, n)

result = calculator.pow()

print(f"{x} raised to the power of {n} is: {result}")
```

OUTPUT:



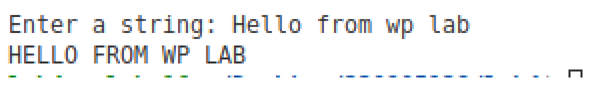
```
Enter the base (x): 4
Enter the exponent (n): 3
4.0 raised to the power of 3 is: 64.0
```

Q7)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 StringProcessor:
def __init__(self):
self.user_string = ""

def get_String(self):
self.user_string = input("Enter a string: ")

def print_String(self):
print(self.user_string.upper())

processor = StringProcessor()
processor.get_String()
processor.print_String()
```

OUTPUT:



```
Enter a string: Hello from wp lab
HELLO FROM WP LAB
```