WPL LAB 4

Name: Abhinav Singh Bhagtana

Section: A

Roll no: 29

Reg no: 230905225

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

```
def reverse_file_content(input_file, output_file, mode='lines'):
  try:
    with open(input_file, 'r') as f:
      content = f.readlines()

    if mode == 'all':
      # Reverse every single character in the file
      reversed_content = "".join(content)[::-1]
    else:
      # Reverse the order of the lines (standard behavior)
      reversed_content = "".join(content[::-1])

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

    print(f"Success! Reverted content saved to {output_file}")

  except FileNotFoundError:
    print("Error: The source file does not exist.")
  except Exception as e:
```

```
    print(f"An error occurred: {e}")


reverse_file_content('input.txt', 'output.txt', mode='all')
```

INPUT:

hi hello

my name is abhinav

OUTPUT:

vanihba si eman ym

olleh ih

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

```
import random as rd


nums = []

for _ in range(10):

nums.append(round(rd.random()*10,2))

nums.sort()

print('nums: ',nums)

print('enter target')

target = float(input())

def binarySearch(left,right):

mid = left + (right-left)//2

if nums[mid] == target:
```

```
return mid

if nums[mid] > target:

return binarySearch(left,mid-1)

elif nums[mid] < target:

return binarySearch(mid+1,right)

return -1

index = binarySearch(0,len(nums))

print(f'found {target} at index: {index}')
```

OUTPUT:

```
nums:  [1.49, 3.07, 4.45, 4.92, 4.98, 6.75, 8.78, 9.33, 9.57, 9.71]

enter target

8.78

found 8.78 at index: 6
```

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

```
words = [input(f"Enter word {i+1}: ") for i in range(5)]


for i in range(len(words)):

for j in range(len(words)):

if words[j] > words[i]:

words[j],words[i]=words[i],words[j]
```

print(words)

OUTPUT:

Enter word 1: zebra

Enter word 2: octopus

Enter word 3: ant

Enter word 4: jellyfish

Enter word 5: dolphin

['ant', 'dolphin', 'jellyfish', 'octopus', 'zebra']

Q4. Write a Python class to get all possible unique subsets from a set of distinct

```
class Subsetter:
    def get_subsets(self, nums):
        result = []
        self._backtrack(sorted(nums), 0, [], result)
        return result

    def _backtrack(self, nums, start, path, result):
        result.append(list(path))

        for i in range(start, len(nums)):
```

```
        path.append(nums[i])

        self._backtrack(nums, i + 1, path, result)

        path.pop()

solver = Subsetter()
input_list = [int(input(f'enter {i+1} num ')) for i in range(4)]
output = solver.get_subsets(input_list)

print(f"Input: {input_list}")
print(f"Output: {output}")
```

OUTPUT:

```
enter 1 num 1
enter 2 num 2
enter 3 num 3
enter 4 num 4
Input: [1, 2, 3, 4]
Output: [[], [1], [1, 2], [1, 2, 3], [1, 2, 3, 4], [1, 2, 4], [1, 3], [1, 3, 4], [1, 4], [2], [2, 3], [2, 3, 4], [2, 4], [3], [3, 4], [4]]
```

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

Output: 3, 4.

```
class findPair:

def find(self,arr,target):

for i in range(len(arr)-1):

for j in range(i+1,len(arr)):

if arr[i] + arr[j] == target:

return i,j

return None

n = int(input('how many numbers '))

numbers = [int(input(f'enter {i+1} num ')) for i in range(n)]

obj = findPair()

target = int(input('enter target '))

i,j = obj.find(numbers,target)

print(i,',',j)
```

OUTPUT:

```
how many numbers 10

enter 1 num 1
```

enter 2 num 2

enter 3 num 3

enter 4 num 4

enter 5 num 5

enter 6 num 6

enter 7 num 7

enter 8 num 8

enter 9 num 9

enter 10 num 11

enter target 11

1 , 8

Q6. Write a Python class to implement pow(x, n).

```
class power:

def pow(self,n,p):

prod = n

for i in range(p):

prod *= n

return prod


obj = power()
```

```
num = int(input('enter number: '))

powr = int(input('enter power: '))

print(f'pow({num},{powr}): ',pow(num,powr))
```

OUTPUT:

```
enter number: 13

enter power: 3

pow(13,3):  2197
```

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.

```
class Strings:

def get_String(self):

self.sent = input('enter string ')

def print_String(self):

print(self.sent.upper())

obj = Strings()

obj.get_String()

obj.print_String()
```

OUTPUT:

enter string hello world

HELLO WORLD