

Lab 4 — Python Basics

Name: Aditya Sinha

Reg. No: 230905218

Class & Section: CSE-A1

Roll No: 27

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

Code :

```
src = open( "file1.txt", "r" )
data = src.read()
src.close()

data = data[::-1]

dst = open( "output1.txt", "w+" )
dst.write(data)
dst.close()
```

Input(file1.txt):

```
hello
```

Output(output1.txt):

```
olleh
```

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

Code :

```
def binSearch( l, x, low, high ):
    if low > high :
        return -1
    mid = low + (high - low)//2
    if l[mid] < x :
        return binSearch( l, x, mid + 1, high )
    elif l[mid] == x :
```

```
        return mid
else :
    return binSearch( l, x, low, mid - 1)

l = [1, 4, 5, 6, 8, 9, 23]
x = 8
y = 139

print("Index of x:", binSearch(l, x, 0, len(l) - 1))
print("Index of y:", binSearch(l, y, 0, len(l) - 1))
```

Output:

```
Index of x: 4
Index of y: -1
```

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

Code :

```
l = ["abc", "zh", "hello"]
l.sort()
print(l)
```

Output:

```
['abc', 'hello', 'zh']
```

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

Code :

```
def takeNotTake(arr, idx, currList, globalList):
    if idx == len(arr):
        globalList.append(currList.copy())
        return

    currList.append(arr[idx])
    takeNotTake(arr, idx + 1, currList, globalList)

    currList.pop()
```

```

takeNotTake(arr, idx + 1, currList, globalList)

arr = [4, 5, 6]
globalList = []

takeNotTake(arr, 0, [], globalList)

print(globalList[::-1])

```

Output:

```
[[], [6], [5], [5, 6], [4], [4, 6], [4, 5], [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 TwoSum:
    def __init__(self, nums, target):
        self.nums = nums
        self.target = target

    def compute(self):
        numToIdx = {}

        for idx, num in enumerate(self.nums, 1):
            complement = self.target - num

            if complement in numToIdx:
                return [numToIdx[complement], idx]

            numToIdx[num] = idx

        return -1

nums = [10, 20, 10, 40, 50, 60, 70]
target = 50

obj = TwoSum(nums, target)
result = obj.compute()

print("Indices of elements whose sum is", target, ":", result)

```

Output:

/

```
Indices of elements whose sum is 50 : [3, 4]
```

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

Code :

```
def pow(x, n):
    if n == 0:
        return 1
    y = pow(x, n//2)*pow(x, n//2)
    if (n&1) == 0:
        return y
    else:
        return x*y

print(pow(5, 3))
```

Output:

```
125
```

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 StringManipulator:
    def __init__(self):
        self.s = ""

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

    def print_String(self):
        print("String in uppercase:", self.s.upper())

obj = StringManipulator()
obj.get_String()
obj.print_String()
```

Output:

```
Enter a string: hello
String in uppercase: HELLO
```