

Lab 4 – Python Basics

Q1.

```
def reverse_file(input_file, output_file):
    try:
        with open(input_file, 'r') as f:
            content = f.read()
            print(f"Original content: {content}")
        with open(output_file, 'w') as f:
            f.write(content[::-1])
        print(f"Content reversed and saved to {output_file}")
    except FileNotFoundError:
        print("Source file not found.")

if __name__ == "__main__":
    with open("in.txt", "w") as f:
        f.write("Hello World")
    reverse_file('in.txt', 'out.txt')
```

Q2.

```
def binary_search(arr, low, high, x):
    if high >= low:
        mid = (high + low) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binary_search(arr, low, mid - 1, x)
        else:
            return binary_search(arr, mid + 1, high, x)
    else:
        return -1

if __name__ == "__main__":
    data = [10, 20, 30, 40, 50]
    target = 20
    print(f"Input Array: {data}")
    print(f"Target: {target}")
    result = binary_search(data, 0, len(data) - 1, target)
    print(f"Element found at index: {result}" if result != -1 else "Not found")
```

Q3.

```
def sort_words():
    sentence = input("Enter words: ")
    words = [word.lower() for word in sentence.split()]
```

```
words.sort()
print("Sorted words:", " ".join(words))
if __name__ == "__main__":
    sort_words()
```

Q4.

```
class Subsets:
    def get_subsets(self, sset):
        return self.backtrack([], sorted(sset))

    def backtrack(self, current, sset):
        if sset:
            return self.backtrack(current, sset[1:]) + self.backtrack(current + [sset[0]], sset[1:])
        return [current]

if __name__ == "__main__":
    ob = Subsets()
    input_set = [1, 2, 3]
    print(f"Input Set: {input_set}")
    print("Subsets:")
    print(ob.get_subsets(input_set))
```

Q5.

```
class TwoSum:
    def find_indices(self, nums, target):
        lookup = {}
        for i, num in enumerate(nums):
            if target - num in lookup:
                return (lookup[target - num], i)
            lookup[num] = i

if __name__ == "__main__":
    numbers = [10, 20, 10, 40, 50, 60, 70]
    print(f"Data: {numbers}")
    target_val = 50
    print(f"Target value: {target_val}")
    print(TwoSum().find_indices(numbers, target_val))
```

Q6.

```
class Power:
    def myPow(self, x, n):
        if n < 0:
            x = 1 / x
            n = -n
        res = 1
```

```
while n:
    if n % 2:
        res *= x
        x *= x
        n //= 2
    return res

if __name__ == "__main__":
    print("Pow(3^5): ",Power().myPow(3, 5))
```

Q7.

```
class StringProcessor:
    def __init__(self):
        self.s = ""

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

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

if __name__ == "__main__":
    obj = StringProcessor()
    obj.get_String()
    obj.print_String()
```

Outputs:

```
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q1.py
Original content: Hello World
Content reversed and saved to out.txt
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ cat out.txt
dlroW olleHWP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q2.py
Input Array: [10, 20, 30, 40, 50]
Target: 20
Element found at index: 1
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q3.py
Enter words: abc xyz pqr
Sorted words: abc pqr xyz
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q4.py
Input Set: [1, 2, 3]
Subsets:
[[], [3], [2], [2, 3], [1], [1, 3], [1, 2], [1, 2, 3]]
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q5.py
Data: [10, 20, 10, 40, 50, 60, 70]
Target value: 50
(2, 3)
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q6.py
Pow(3^5): 243
WP_C1@CL3-23:~/Documents/230905152_WP/L4$ python3 q7.py
Enter a string: webdev
WEBDEV
WP_C1@CL3-23:~/Documents/230905152_WP/L4$
```
