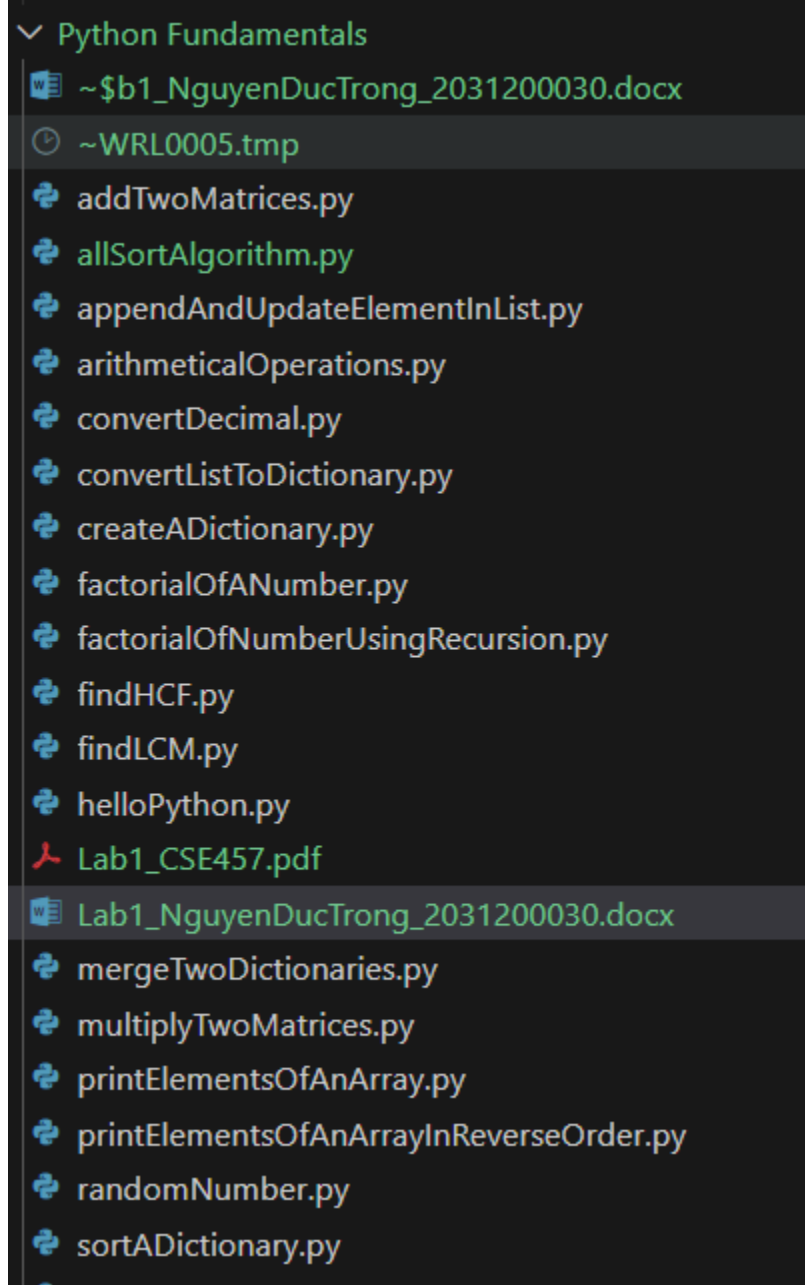
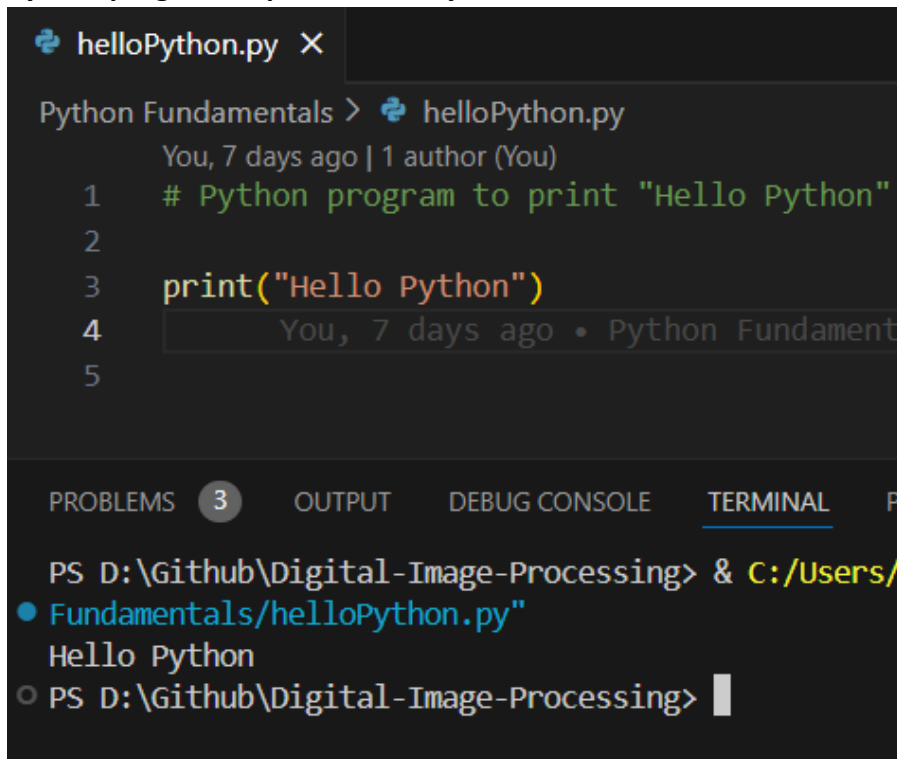


## List of Python Programs for Assignment 1



## 1. Python program to print "Hello Python"



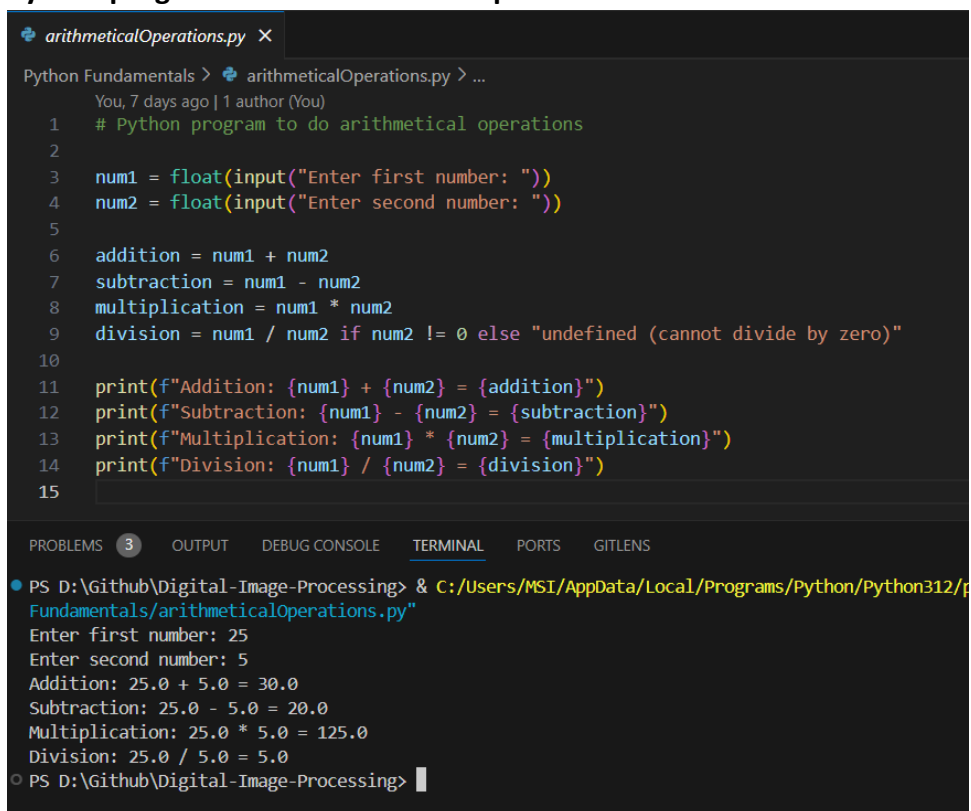
The screenshot shows a code editor with a file named `helloPython.py`. The code is as follows:

```
1 # Python program to print "Hello Python"
2
3 print("Hello Python")
4
5
```

Below the code editor, the terminal output is shown:

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/Python.exe Fundamentals/helloPython.py
Hello Python
PS D:\Github\Digital-Image-Processing>
```

## 2. Python program to do arithmetical operations



The screenshot shows a code editor with a file named `arithmeticalOperations.py`. The code is as follows:

```
1 # Python program to do arithmetical operations
2
3 num1 = float(input("Enter first number: "))
4 num2 = float(input("Enter second number: "))
5
6 addition = num1 + num2
7 subtraction = num1 - num2
8 multiplication = num1 * num2
9 division = num1 / num2 if num2 != 0 else "undefined (cannot divide by zero)"
10
11 print(f"Addition: {num1} + {num2} = {addition}")
12 print(f"Subtraction: {num1} - {num2} = {subtraction}")
13 print(f"Multiplication: {num1} * {num2} = {multiplication}")
14 print(f"Division: {num1} / {num2} = {division}")
15
```

Below the code editor, the terminal output is shown:

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/Python.exe Fundamentals/arithmeticalOperations.py
Enter first number: 25
Enter second number: 5
Addition: 25.0 + 5.0 = 30.0
Subtraction: 25.0 - 5.0 = 20.0
Multiplication: 25.0 * 5.0 = 125.0
Division: 25.0 / 5.0 = 5.0
PS D:\Github\Digital-Image-Processing>
```

### 3. Python program to find the area of a triangle

```
theAreaOfATriangle.py ×
Python Fundamentals > theAreaOfATriangle.py > herons_formula

5 def right_angled_triangle():
6     base = float(input("Enter the base of the triangle: "))
7     height = float(input("Enter the height of the triangle: "))
8
9     area = 0.5 * base * height
10
11     print(f"The area of the triangle with base {base} and height {height} is: {area}")
12
13 right_angled_triangle()
14
15 # Area of a Triangle in Python Using Heron's Formula
16 def herons_formula():
17     side1 = float(input("Enter the first side of the triangle: "))
18     side2 = float(input("Enter the second side of the triangle: "))
19     side3 = float(input("Enter the third side of the triangle: "))
20
21     s = (side1 + side2 + side3) / 2
22     area = (s*(s-side1)*(s-side2)*(s-side3))**0.5
23
24     print(f"The area of the triangle with Heron's Formula is: {area}")
25
26     # format in two decimal digits
27     print('The area of the triangle with Heron's Formula is: {:.2f}'.format(area))
28
29 herons_formula()
```

PROBLEMS 3 OUTPUT DEBUG CONSOLE TERMINAL PORTS GITLENS

```
Enter the base of the triangle: 5
Enter the height of the triangle: 3
The area of the triangle with base 5.0 and height 3.0 is: 7.5
Enter the first side of the triangle: 3
Enter the second side of the triangle: 4
Enter the third side of the triangle: 5
The area of the triangle with Heron's Formula is: 6.0
The area of the triangle with Heron's Formula is: 6.00
PS D:\Github\Digital-Image-Processing>
```

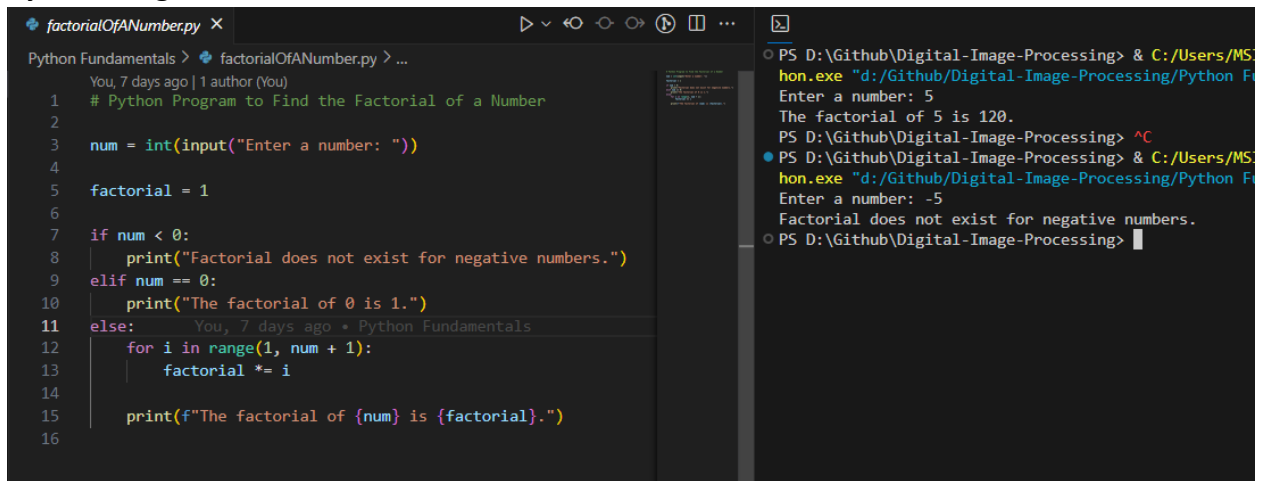
#### 4. Python program to generate a random number

```
randomNumber.py ×
Python Fundamentals > randomNumber.py > ...

3 import random
4
5 num = random.random()
6 print(num)
7
8 # Generating a random number between 1 and 100
9 random_number = random.randint(1, 100)
10 print(random_number)
11
12 # Generating a Random number using choice()
13 # prints a random value from the list
14 list1 = [1, 2, 3, 4, 5, 6, 7, 8, 9]
15 print(random.choice(list1))
16 # prints a random item from the string
17 string = "striver"
18 print(random.choice(string))
19
20 # Generating a Random Number using randrange()
21 # using randrange() to generate in range from 20 to 50
22 print("A random number from range is : ", end="")
23 print(random.randrange(20, 80, 5))
24
25 # Generating a Random number using shuffle()
26 # declare a list
27 sample_list = ['A', 'B', 'C', 'D', 'E']
28
29 print("Original list : ")
30 print(sample_list)
31
32 # first shuffle
33 random.shuffle(sample_list)
34 print("\nAfter the first shuffle : ")
35 print(sample_list)
36
37 # second shuffle
38 random.shuffle(sample_list)
39 print("\nAfter the second shuffle : ")
40 print(sample_list)
```

PS D:\Github\Digital-Image-Processing> &  
o :./Github/Digital-Image-Processing/Python  
0.3825776111827782  
66  
7  
r  
A random number from range is : 35  
Original list :  
['A', 'B', 'C', 'D', 'E']  
  
After the first shuffle :  
['A', 'D', 'B', 'C', 'E']  
  
After the second shuffle :  
['C', 'D', 'A', 'B', 'E']  
PS D:\Github\Digital-Image-Processing>

## 5. Python Program to Find the Factorial of a Number

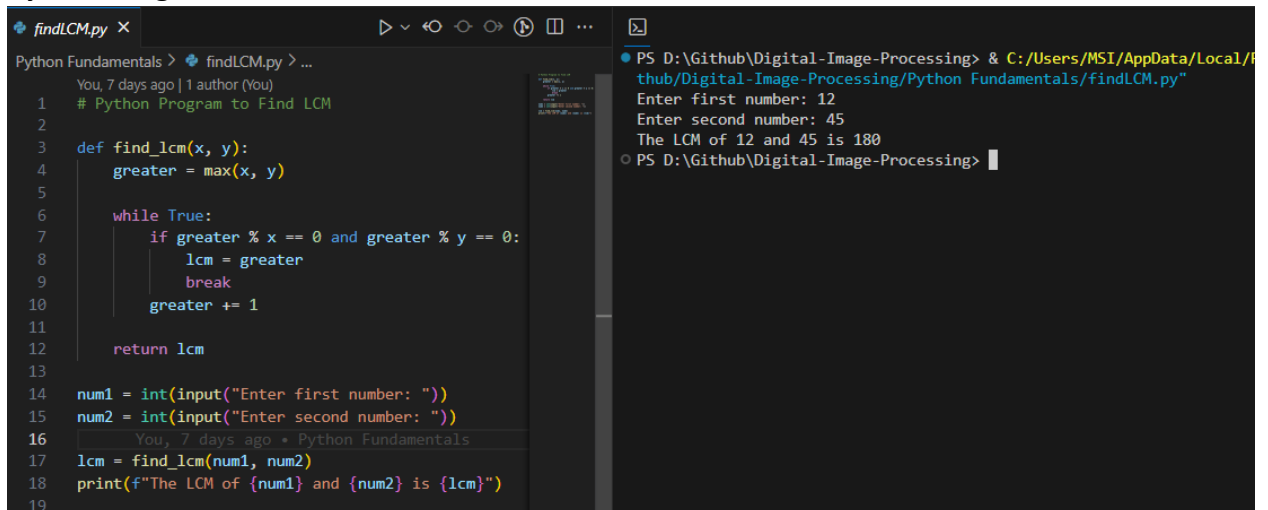


The screenshot shows a Python IDE with a file named `factorialOfANumber.py`. The code is as follows:

```
1 # Python Program to Find the Factorial of a Number
2
3 num = int(input("Enter a number: "))
4
5 factorial = 1
6
7 if num < 0:
8     print("Factorial does not exist for negative numbers.")
9 elif num == 0:
10    print("The factorial of 0 is 1.")
11 else:
12    for i in range(1, num + 1):
13        factorial *= i
14
15    print(f"The factorial of {num} is {factorial}.")
16
```

The terminal output shows the program being executed twice. First, the user enters 5, and the program outputs "The factorial of 5 is 120." Then, the user enters -5, and the program outputs "Factorial does not exist for negative numbers."

## 6. Python Program to Find LCM

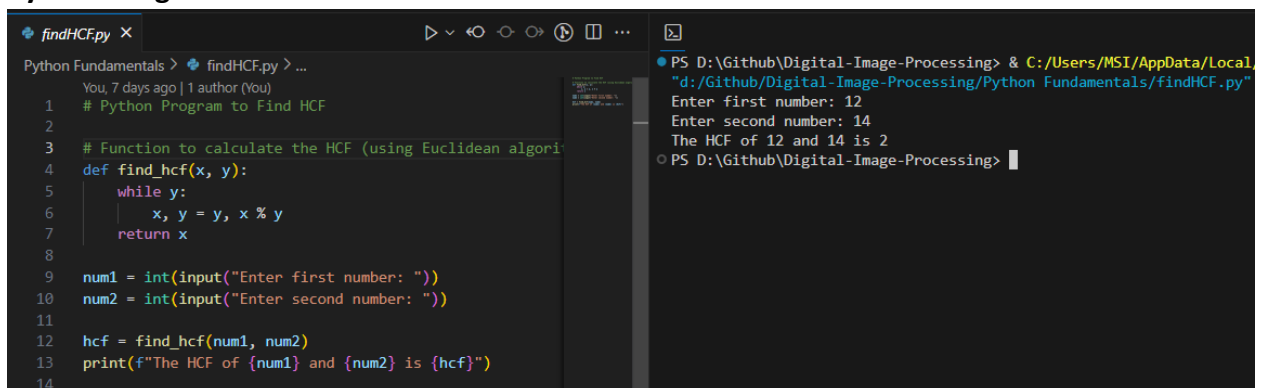


The screenshot shows a Python IDE with a file named `findLCM.py`. The code is as follows:

```
1 # Python Program to Find LCM
2
3 def find_lcm(x, y):
4     greater = max(x, y)
5
6     while True:
7         if greater % x == 0 and greater % y == 0:
8             lcm = greater
9             break
10        greater += 1
11
12    return lcm
13
14 num1 = int(input("Enter first number: "))
15 num2 = int(input("Enter second number: "))
16
17 lcm = find_lcm(num1, num2)
18 print(f"The LCM of {num1} and {num2} is {lcm}")
19
```

The terminal output shows the program being executed once. The user enters 12 for the first number and 45 for the second number, and the program outputs "The LCM of 12 and 45 is 180."

## 7. Python Program to Find HCF



The screenshot shows a Python IDE with a file named `findHCF.py`. The code is as follows:

```
1 # Python Program to Find HCF
2
3 # Function to calculate the HCF (using Euclidean algorithm)
4 def find_hcf(x, y):
5     while y:
6         x, y = y, x % y
7     return x
8
9 num1 = int(input("Enter first number: "))
10 num2 = int(input("Enter second number: "))
11
12 hcf = find_hcf(num1, num2)
13 print(f"The HCF of {num1} and {num2} is {hcf}")
14
```

The terminal output shows the program being executed once. The user enters 12 for the first number and 14 for the second number, and the program outputs "The HCF of 12 and 14 is 2."

## 8. Python Program to Convert Decimal to Binary, Octal and Hexadecimal

```
convertDecimal.py X
Python Fundamentals > convertDecimal.py > ...
You, 7 days ago | 1 author (You)
1 # Python Program to Convert Decimal to Binary, Octal and Hexadecimal
2
3 decimal_num = int(input("Enter a decimal number: "))
4
5 binary_num = bin(decimal_num)
6 octal_num = oct(decimal_num)
7 hexadecimal_num = hex(decimal_num)
8
9 print(f"Binary of {decimal_num} is {binary_num[2:]}")
10 print(f"Octal of {decimal_num} is {octal_num[2:]}")
11 print(f"Hexadecimal of {decimal_num} is {hexadecimal_num[2:]}")
12
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/PythonFundamentals/convertDecimal.py"

Enter a decimal number: 123  
Binary of 123 is 1111011  
Octal of 123 is 173  
Hexadecimal of 123 is 7b  
PS D:\Github\Digital-Image-Processing>

## 9. Python Program to Find Factorial of Number Using Recursion

```
factorialOfNumberUsingRecursion.py X
Python Fundamentals > factorialOfNumberUsingRecursion.py > ...
You, 7 days ago | 1 author (You)
1 # Python Program to Find Factorial of Number Using Recursion
2
3 def factorial(n):
4     if n == 0 or n == 1:
5         return 1
6     else:
7         return n * factorial(n - 1) # Recursive case
8
9 num = int(input("Enter a number: "))
10
11 if num < 0:
12     print("Factorial does not exist for negative numbers.")
13 else:
14     print(f"The factorial of {num} is {factorial(num)}.")
15
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/PythonFundamentals/factorialOfNumberUsingRecursion.py"

Enter a number: 7  
The factorial of 7 is 5040.  
PS D:\Github\Digital-Image-Processing>

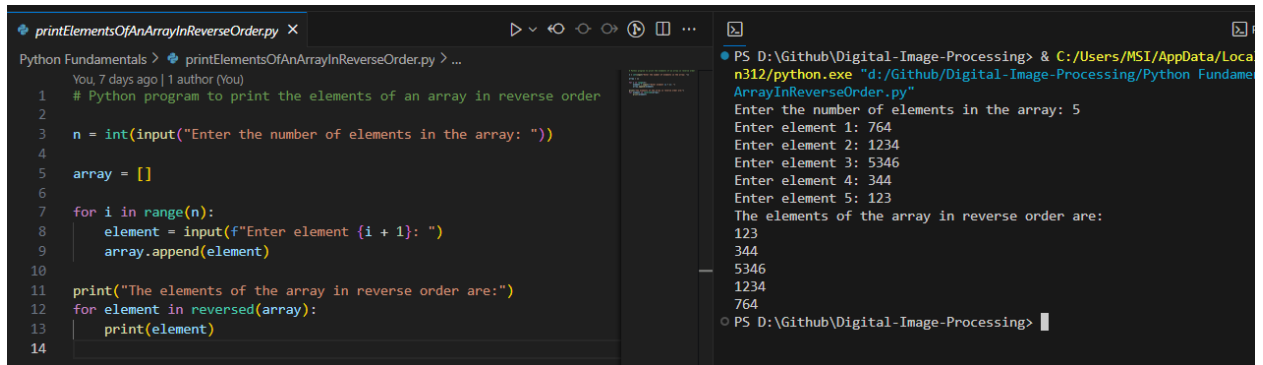
## 10. Python program to print the elements of an array

```
printElementsOfAnArray.py X
Python Fundamentals > printElementsOfAnArray.py > ...
You, 7 days ago | 1 author (You)
1 # Python program to print the elements of an array
2
3 n = int(input("Enter the number of elements in the array: "))
4
5 array = []
6
7 for i in range(n):
8     element = input(f"Enter element {i + 1}: ")
9     array.append(element)
10
11 print("The elements of the array are:")
12 for element in array:
13     print(element)
14
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/PythonFundamentals/printElementsOfAnArray.py"

Enter the number of elements in the array: 5  
Enter element 1: 213  
Enter element 2: 745  
Enter element 3: 4132  
Enter element 4: 34  
Enter element 5: 6  
The elements of the array are:  
213  
745  
4132  
34  
6  
PS D:\Github\Digital-Image-Processing>

## 11. Python program to print the elements of an array in reverse order



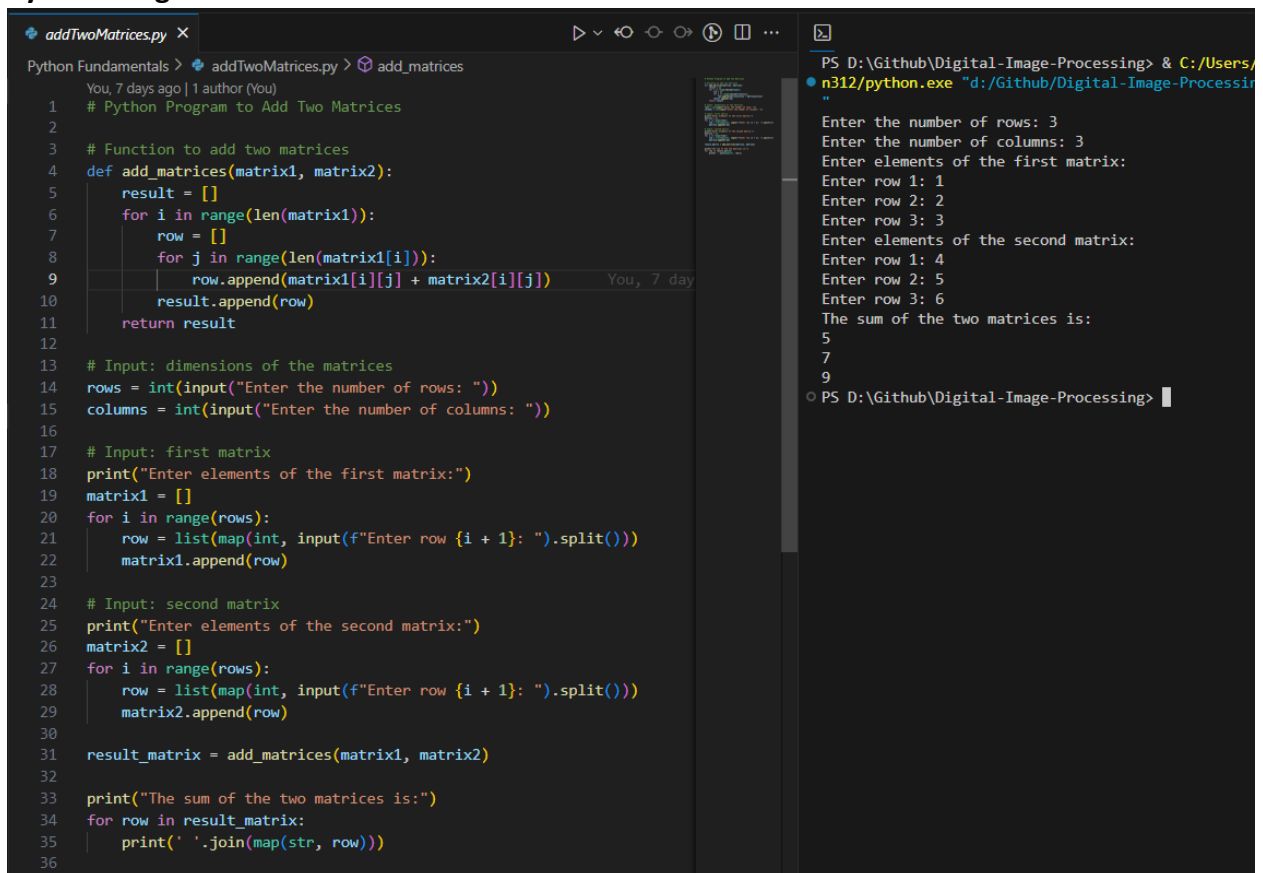
The screenshot shows a Python IDE with a file named `printElementsOfAnArrayInReverseOrder.py`. The code is as follows:

```
1 # Python program to print the elements of an array in reverse order
2
3 n = int(input("Enter the number of elements in the array: "))
4
5 array = []
6
7 for i in range(n):
8     element = input(f"Enter element {i + 1}: ")
9     array.append(element)
10
11 print("The elements of the array in reverse order are:")
12 for element in reversed(array):
13     print(element)
14
```

The terminal output on the right shows the execution of the program:

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/ArrayInReverseOrder.py"
Enter the number of elements in the array: 5
Enter element 1: 764
Enter element 2: 1234
Enter element 3: 5346
Enter element 4: 344
Enter element 5: 123
The elements of the array in reverse order are:
123
344
5346
1234
764
PS D:\Github\Digital-Image-Processing>
```

## 12. Python Program to Add Two Matrices



The screenshot shows a Python IDE with a file named `addTwoMatrices.py`. The code is as follows:

```
1 # Python Program to Add Two Matrices
2
3 # Function to add two matrices
4 def add_matrices(matrix1, matrix2):
5     result = []
6     for i in range(len(matrix1)):
7         row = []
8         for j in range(len(matrix1[i])):
9             row.append(matrix1[i][j] + matrix2[i][j])
10        result.append(row)
11    return result
12
13 # Input: dimensions of the matrices
14 rows = int(input("Enter the number of rows: "))
15 columns = int(input("Enter the number of columns: "))
16
17 # Input: first matrix
18 print("Enter elements of the first matrix:")
19 matrix1 = []
20 for i in range(rows):
21     row = list(map(int, input(f"Enter row {i + 1}: ").split()))
22     matrix1.append(row)
23
24 # Input: second matrix
25 print("Enter elements of the second matrix:")
26 matrix2 = []
27 for i in range(rows):
28     row = list(map(int, input(f"Enter row {i + 1}: ").split()))
29     matrix2.append(row)
30
31 result_matrix = add_matrices(matrix1, matrix2)
32
33 print("The sum of the two matrices is:")
34 for row in result_matrix:
35     print(' '.join(map(str, row)))
36
```

The terminal output on the right shows the execution of the program:

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/AddTwoMatrices.py"
Enter the number of rows: 3
Enter the number of columns: 3
Enter elements of the first matrix:
Enter row 1: 1
Enter row 2: 2
Enter row 3: 3
Enter elements of the second matrix:
Enter row 1: 4
Enter row 2: 5
Enter row 3: 6
The sum of the two matrices is:
5
7
9
PS D:\Github\Digital-Image-Processing>
```

### 13. Python Program to Multiply Two Matrices

```
multiplyTwoMatrices.py x
Python Fundamentals > multiplyTwoMatrices.py > multiply_matrices
You 7 days ago | 1 author (You)
# Python Program to Multiply Two Matrices

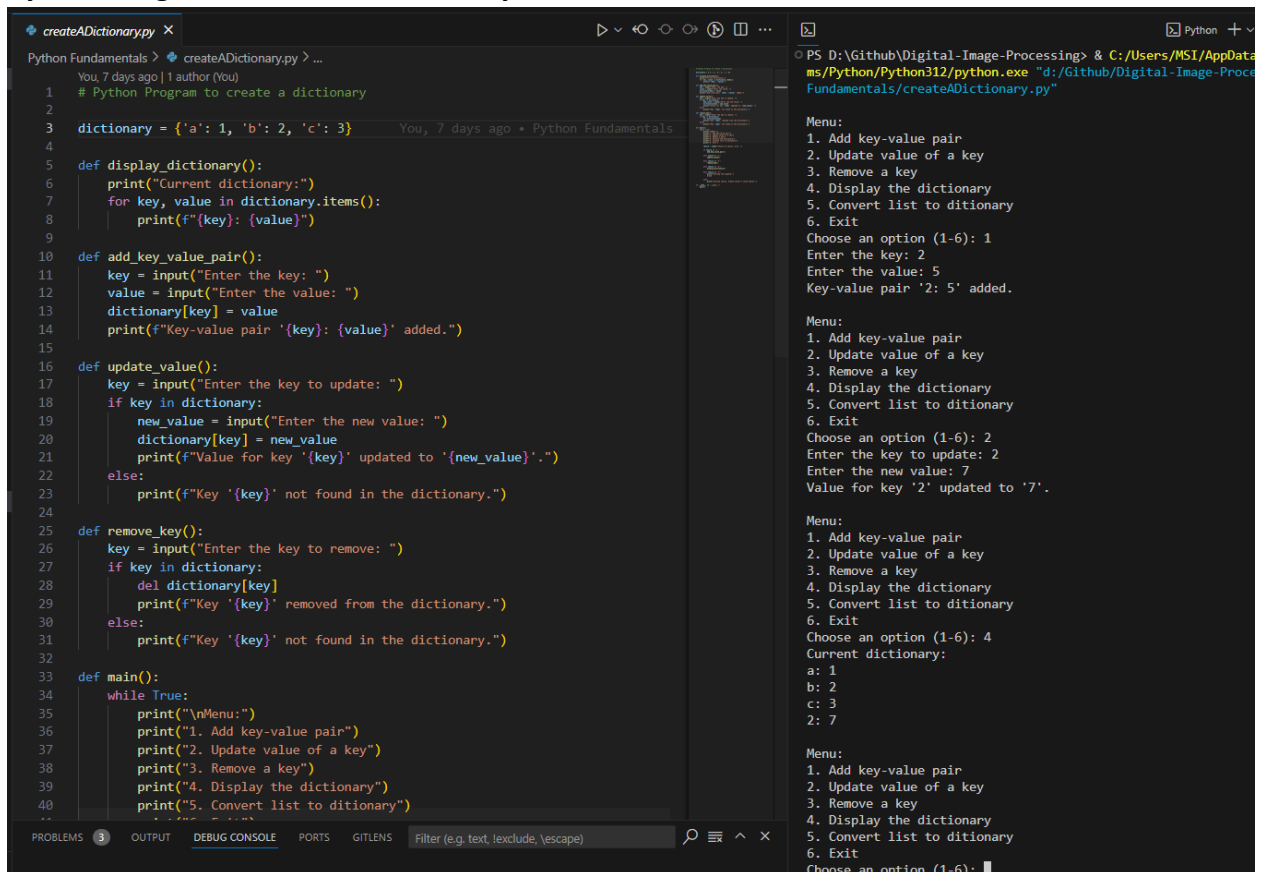
1
2
3 # Function to multiply two matrices
4 def multiply_matrices(matrix1, matrix2):
5     # Get the number of rows and columns
6     rows1 = len(matrix1)
7     cols1 = len(matrix1[0])
8     rows2 = len(matrix2)
9     cols2 = len(matrix2[0])
10
11     You 7 days ago • Python Fundamentals
12     # Initialize the result matrix with zeros
13     result = [[0 for _ in range(cols2)] for _ in range(rows1)]
14
15     # Check if multiplication is possible
16     if cols1 != rows2:
17         print("Matrix multiplication is not possible due to incompatible dimensions.")
18         return None
19
20     # Performing matrix multiplication
21     for i in range(rows1):
22         for j in range(cols2):
23             for k in range(cols1):
24                 result[i][j] += matrix1[i][k] * matrix2[k][j]
25
26     return result
27
28 # Input: dimensions of the matrices
29 rows1 = int(input("Enter the number of rows for the first matrix: "))
30 cols1 = int(input("Enter the number of columns for the first matrix: "))
31 rows2 = int(input("Enter the number of rows for the second matrix: "))
32 cols2 = int(input("Enter the number of columns for the second matrix: "))
33
34 # Input: first matrix
35 print("Enter elements of the first matrix:")
36 matrix1 = []
37 for i in range(rows1):
38     row = list(map(int, input(f"Enter row {i + 1}: ").split()))
39     matrix1.append(row)
40
41 # Input: second matrix
42 print("Enter elements of the second matrix:")
43 matrix2 = []
44 for i in range(rows2):
45     row = list(map(int, input(f"Enter row {i + 1}: ").split()))
46     matrix2.append(row)
47
48 # Matrix multiplication
49 result = multiply_matrices(matrix1, matrix2)
50
51 if result is None:
52     print("Matrix multiplication is not possible due to incompatible dimensions.")
53 else:
54     print("Result matrix:")
55     for i in range(len(result)):
56         for j in range(len(result[0])):
57             print(result[i][j], end=" ")
58         print()
59
60 PS D:\Github\Digital-Image-Processing> C:\Users\MSI\AppData\Local\Programs\Python\Python312\python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/multiplyTwoMatrices.py"
Enter the number of rows for the first matrix: 3
Enter the number of columns for the first matrix: 2
Enter the number of rows for the second matrix: 3
Enter the number of columns for the second matrix: 2
Enter elements of the first matrix:
Enter row 1: 1
Enter row 2: 2
Enter row 3: 3
Enter elements of the second matrix:
Enter row 1: 1
Enter row 2: 2
Enter row 3: 3
Matrix multiplication is not possible due to incompatible dimensions.
PS D:\Github\Digital-Image-Processing>
```



#### 14. Python Program to append element in the list and update list with insertion of elements, removing an element, comparison of two lists, etc.

```
appendAndUpdateElementInList.py X
Python Fundamentals > appendAndUpdateElementInList.py > main
You, 7 days ago | 1 author (You)
1 # Python Program to append element in the list and update list with insertion of e
2 # removing an element, comparison of two lists, etc
3 my_list = []
4
5 def display_list(list):
6     print("Current list:", list)
7
8 def append_element():
9     element = input("Enter an element to append to the list: ")
10    my_list.append(element)
11    print(f"Element '{element}' appended.")
12
13 def insert_element():
14    position = int(input("Enter the position (index) to insert an element (0-based)
15    element_to_insert = input("Enter the element to insert: ")
16    if 0 <= position <= len(my_list):
17        my_list.insert(position, element_to_insert)
18        print(f"Element '{element_to_insert}' inserted at position {position}.")
19    else:
20        print("Invalid position.")
21
22 def remove_element():
23    element_to_remove = input("Enter the element to remove from the list: ")
24    if element_to_remove in my_list:
25        my_list.remove(element_to_remove)
26        print(f"Element '{element_to_remove}' removed.")
27    else:
28        print(f"Element '{element_to_remove}' not found in the list.")
29
30 def enter_another_list():
31    another_list = input("Enter another list of elements separated by space: ").sp
32    print("List 1:", my_list)
33    print("List 2:", another_list)
34    if my_list == another_list:
35        print("Both lists are equal.")
36    else:
37        print("The lists are not equal.")
38
39 def main():
40    while True:
        Menu:
        1. Append element to the list
        2. Insert element at a specific position
        3. Remove an element from the list
        4. Compare two lists
        5. Display the current list
        6. Exit
        Choose an option (1-6): 1
        Enter an element to append to the list: 12
        Element '12' appended.
        Menu:
        1. Append element to the list
        2. Insert element at a specific position
        3. Remove an element from the list
        4. Compare two lists
        5. Display the current list
        6. Exit
        Choose an option (1-6): 5
        Current list: ['12']
        Menu:
        1. Append element to the list
        2. Insert element at a specific position
        3. Remove an element from the list
        4. Compare two lists
        5. Display the current list
        6. Exit
        Choose an option (1-6): 4
        Enter another list of elements separated by space: 14
        List 1: ['12']
        List 2: ['14']
        The lists are not equal.
        Menu:
        1. Append element to the list
        2. Insert element at a specific position
        3. Remove an element from the list
        4. Compare two lists
        5. Display the current list
        6. Exit
        Choose an option (1-6):
```

## 15. Python Program to create a dictionary



```
createADictionary.py X
Python Fundamentals > createADictionary.py > ...
You, 7 days ago | 1 author (You)
# Python Program to create a dictionary

1 dictionary = {'a': 1, 'b': 2, 'c': 3}
2
3
4
5 def display_dictionary():
6     print("Current dictionary:")
7     for key, value in dictionary.items():
8         print(f"{key}: {value}")
9
10 def add_key_value_pair():
11     key = input("Enter the key: ")
12     value = input("Enter the value: ")
13     dictionary[key] = value
14     print(f"Key-value pair '{key}: {value}' added.")
15
16 def update_value():
17     key = input("Enter the key to update: ")
18     if key in dictionary:
19         new_value = input("Enter the new value: ")
20         dictionary[key] = new_value
21         print(f"Value for key '{key}' updated to '{new_value}'.")
22     else:
23         print(f"Key '{key}' not found in the dictionary.")
24
25 def remove_key():
26     key = input("Enter the key to remove: ")
27     if key in dictionary:
28         del dictionary[key]
29         print(f"Key '{key}' removed from the dictionary.")
30     else:
31         print(f"Key '{key}' not found in the dictionary.")
32
33 def main():
34     while True:
35         print("\nMenu:")
36         print("1. Add key-value pair")
37         print("2. Update value of a key")
38         print("3. Remove a key")
39         print("4. Display the dictionary")
40         print("5. Convert list to dictionary")
41         choice = input("Choose an option (1-6): ")
42         if choice == '1':
43             add_key_value_pair()
44         elif choice == '2':
45             update_value()
46         elif choice == '3':
47             remove_key()
48         elif choice == '4':
49             display_dictionary()
50         elif choice == '5':
51             # Convert list to dictionary
52             list_data = input("Enter list elements separated by space: ")
53             list_data = list_data.split()
54             res_dict = {}
55             for i in range(0, len(list_data), 2):
56                 res_dict[list_data[i]] = list_data[i + 1]
57             print(res_dict)
58         else:
59             print("Invalid choice. Please choose an option from 1 to 6.")
60         print("\nCurrent dictionary:")
61         display_dictionary()
62
63 if __name__ == '__main__':
64     main()
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/ms/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Fundamentals/createADictionary.py"

Menu:

1. Add key-value pair
2. Update value of a key
3. Remove a key
4. Display the dictionary
5. Convert list to dictionary
6. Exit

Choose an option (1-6): 1

Enter the key: 2

Enter the value: 5

Key-value pair '2: 5' added.

Menu:

1. Add key-value pair
2. Update value of a key
3. Remove a key
4. Display the dictionary
5. Convert list to dictionary
6. Exit

Choose an option (1-6): 2

Enter the key to update: 2

Enter the new value: 7

Value for key '2' updated to '7'.

Menu:

1. Add key-value pair
2. Update value of a key
3. Remove a key
4. Display the dictionary
5. Convert list to dictionary
6. Exit

Choose an option (1-6): 4

Current dictionary:

a: 1

b: 2

c: 3

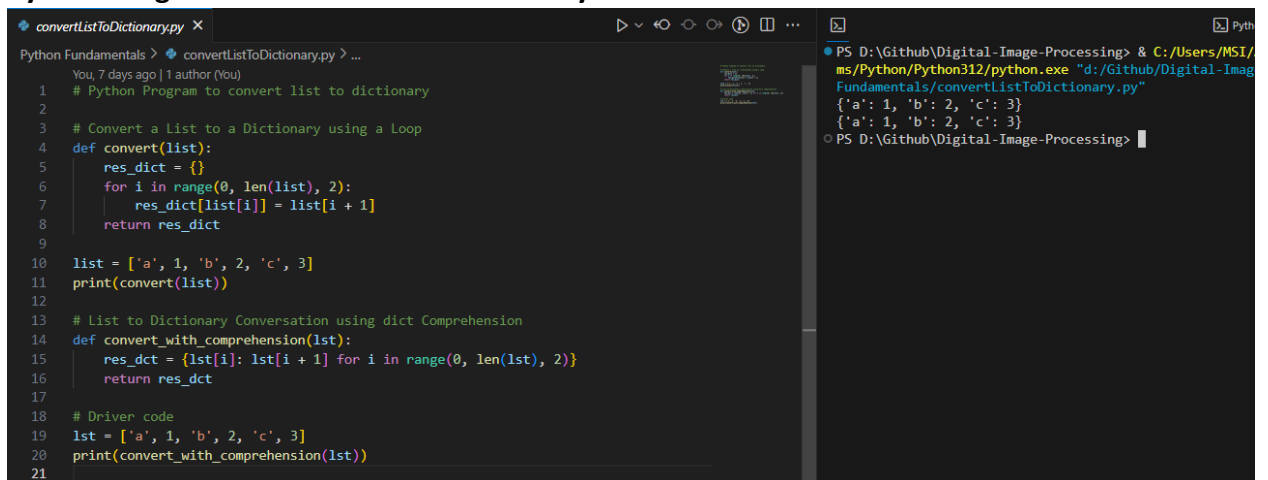
2: 7

Menu:

1. Add key-value pair
2. Update value of a key
3. Remove a key
4. Display the dictionary
5. Convert list to dictionary
6. Exit

Choose an option (1-6):

## 16. Python Program to convert list to dictionary



```
convertListToDictionary.py X
Python Fundamentals > convertListToDictionary.py > ...
You, 7 days ago | 1 author (You)
# Python Program to convert list to dictionary
# Convert a List to a Dictionary using a Loop

1 def convert(list):
2     res_dict = {}
3     for i in range(0, len(list), 2):
4         res_dict[list[i]] = list[i + 1]
5     return res_dict
6
7 list = ['a', 1, 'b', 2, 'c', 3]
8 print(convert(list))
9
10 # List to Dictionary Conversion using dict Comprehension
11 def convert_with_comprehension(lst):
12     res_dct = {lst[i]: lst[i + 1] for i in range(0, len(lst), 2)}
13     return res_dct
14
15 # Driven code
16 lst = ['a', 1, 'b', 2, 'c', 3]
17 print(convert_with_comprehension(lst))
18
19
20
21
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/ms/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Fundamentals/convertListToDictionary.py"

{'a': 1, 'b': 2, 'c': 3}

{'a': 1, 'b': 2, 'c': 3}

PS D:\Github\Digital-Image-Processing>

## 17. Python Program to sort a dictionary

```
sortADictionary.py X
Python Fundamentals > sortADictionary.py > add_key_value_pair
You, 7 days ago | 1 author (You)
my_dict = {'a': 1, 'b': 2, 'c': 3}

1
2
3 def display_dictionary():
4     if not my_dict:
5         print("The dictionary is empty.")
6     else:
7         print("Current dictionary:")
8         for key, value in my_dict.items():
9             print(f"{key}: {value}")
10
11 def add_key_value_pair():
12     key = input("Enter the key: ")
13     value = input("Enter the value: ")
14     my_dict[key] = value
15     print(f"Key-value pair '{key}': {value} added.")
16
17 def sort_dictionary_by_keys():
18     if not my_dict:
19         print("The dictionary is empty. Please add elements to the dictionary first.")
20         return
21
22     sorted_dict = dict(sorted(my_dict.items()))
23     print("Dictionary sorted by keys:", sorted_dict)
24
25 def sort_dictionary_by_values():
26     if not my_dict:
27         print("The dictionary is empty. Please add elements to the dictionary first.")
28         return
29
30     # Convert all values to strings for sorting
31     sorted_dict = dict(sorted(my_dict.items(), key=lambda item: str(item[1])))
32     print("Dictionary sorted by values:", sorted_dict)
33
34
35 def main():
36     while True:
37         print("\nMenu:")
38         print("1. Add a key-value pair")
39         print("2. Display the current dictionary")
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/sortADictionary.py"

Menu:

1. Add a key-value pair
2. Display the current dictionary
3. Sort the dictionary by keys
4. Sort the dictionary by values
5. Exit

Choose an option (1-5): 4  
Dictionary sorted by values: {'a': 1, 'b': 2, 'c': 3}

Menu:

1. Add a key-value pair
2. Display the current dictionary
3. Sort the dictionary by keys
4. Sort the dictionary by values
5. Exit

Choose an option (1-5): 1  
Enter the key:

## 18. Python Program to Merge two Dictionaries

```
mergeTwoDictionaries.py X
Python Fundamentals > mergeTwoDictionaries.py > merge_dictionaries
You, 7 days ago | 1 author (You)
# Python Program to Merge two Dictionaries

1
2
3 dict1 = {'a': 1, 'b': 2, 'c': 3}
4 dict2 = {'a': 11, 'b': 22, 'c': 33}
5
6 def display_dictionaries():
7     print("\nCurrent Dictionaries:")
8     print("Dictionary 1:", dict1)
9     print("Dictionary 2:", dict2)
10
11 def add_key_value_pairs(dict_number):
12     if dict_number == 1:
13         target_dict = dict1
14     else:
15         target_dict = dict2
16
17     while True:
18         key = input(f"Enter a key to add to Dictionary {dict_number} (or type 'done' to finish): ")
19         if key.lower() == 'done':
20             break
21         value = input("Enter the value for the key: ")
22         target_dict[key] = value
23         print(f"Key-value pair '{key}': {value} added to Dictionary {dict_number}.")
24
25 def merge_dictionaries():
26     global dict1, dict2
27     merged_dict = {**dict1, **dict2}
28     print("Merged Dictionary:", merged_dict)
29
30 def main():
31     while True:
32         print("\nMenu:")
33         print("1. Add key-value pairs to Dictionary 1")
34         print("2. Add key-value pairs to Dictionary 2")
35         print("3. Display the current dictionaries")
36         print("4. Merge the two dictionaries")
37         print("5. Exit")
38
39         choice = input("Choose an option (1-5): ")
```

PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/mergeTwoDictionaries.py"

Menu:

1. Add key-value pairs to Dictionary 1
2. Add key-value pairs to Dictionary 2
3. Display the current dictionaries
4. Merge the two dictionaries
5. Exit

Choose an option (1-5): 1  
Enter a key to add to Dictionary 1 (or type 'done' to finish): 2  
Enter the value for the key: 4  
Key-value pair '2': 4 added to Dictionary 1.  
Enter a key to add to Dictionary 1 (or type 'done' to finish): done

Menu:

1. Add key-value pairs to Dictionary 1
2. Add key-value pairs to Dictionary 2
3. Display the current dictionaries
4. Merge the two dictionaries
5. Exit

Choose an option (1-5): 2  
Enter a key to add to Dictionary 2 (or type 'done' to finish): 1  
Enter the value for the key: 4  
Key-value pair '1': 4 added to Dictionary 2.  
Enter a key to add to Dictionary 2 (or type 'done' to finish): done

Menu:

1. Add key-value pairs to Dictionary 1
2. Add key-value pairs to Dictionary 2
3. Display the current dictionaries
4. Merge the two dictionaries
5. Exit

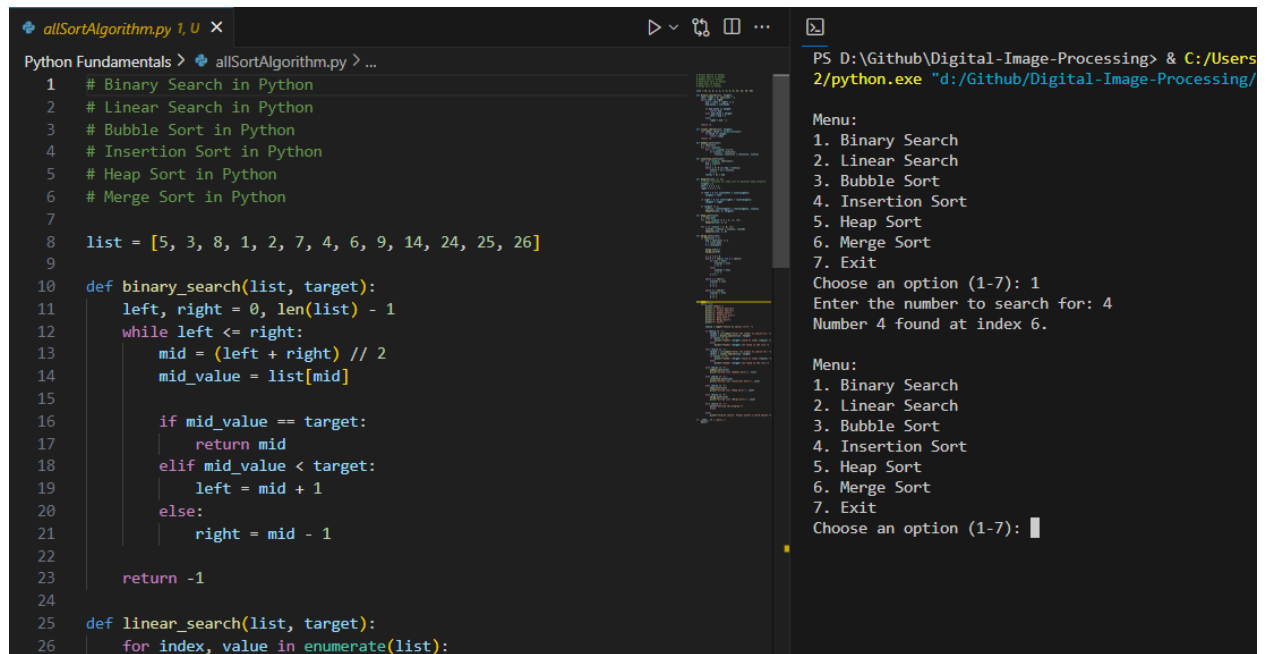
Choose an option (1-5): 4  
Merged Dictionary: {'a': 11, 'b': 22, 'c': 33, '2': '4', '1': '4'}

Menu:

1. Add key-value pairs to Dictionary 1
2. Add key-value pairs to Dictionary 2
3. Display the current dictionaries
4. Merge the two dictionaries
5. Exit

Choose an option (1-5):

## 19. Binary Search in Python



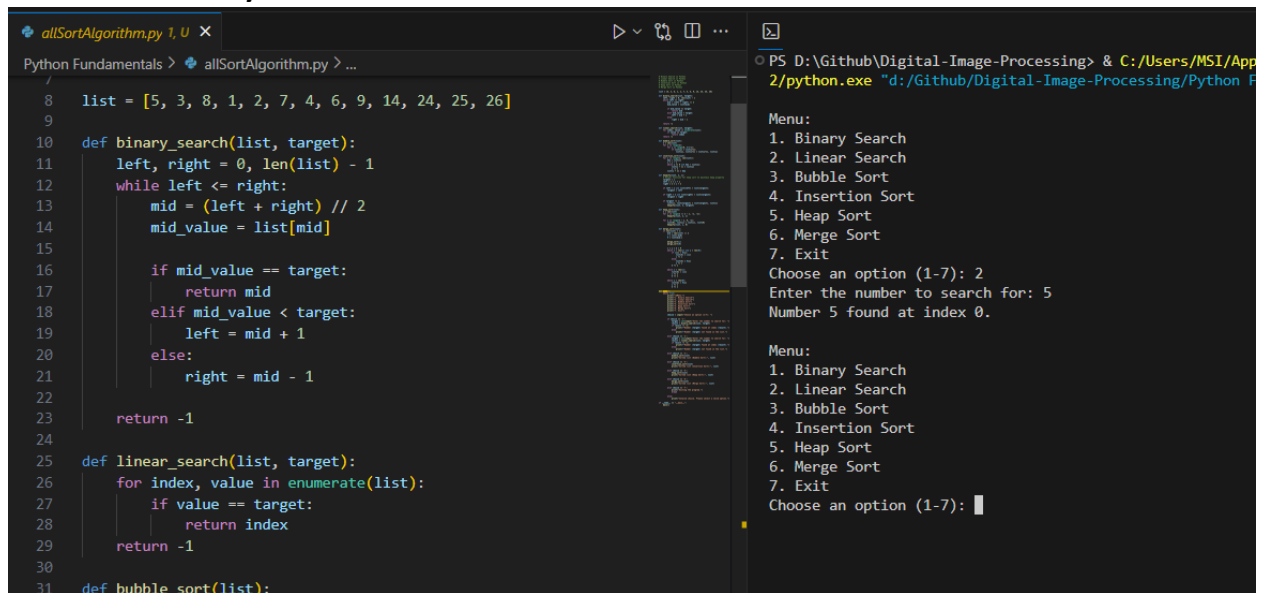
The screenshot shows a Python IDE with a file named `allSortAlgorithm.py`. The code implements a binary search function and a menu-driven interface. The list being searched is `[5, 3, 8, 1, 2, 7, 4, 6, 9, 14, 24, 25, 26]`. The binary search function is defined as follows:

```
def binary_search(list, target):  
    left, right = 0, len(list) - 1  
    while left <= right:  
        mid = (left + right) // 2  
        mid_value = list[mid]  
  
        if mid_value == target:  
            return mid  
        elif mid_value < target:  
            left = mid + 1  
        else:  
            right = mid - 1  
  
    return -1
```

The execution output shows the menu being displayed, the user selecting option 1 (Binary Search), entering the number 4 to search for, and the program returning the index 6.

```
PS D:\Github\Digital-Image-Processing> & C:/Users/2/python.exe "d:/Github/Digital-Image-Processing/  
Menu:  
1. Binary Search  
2. Linear Search  
3. Bubble Sort  
4. Insertion Sort  
5. Heap Sort  
6. Merge Sort  
7. Exit  
Choose an option (1-7): 1  
Enter the number to search for: 4  
Number 4 found at index 6.  
  
Menu:  
1. Binary Search  
2. Linear Search  
3. Bubble Sort  
4. Insertion Sort  
5. Heap Sort  
6. Merge Sort  
7. Exit  
Choose an option (1-7):
```

## 20. Linear Search in Python



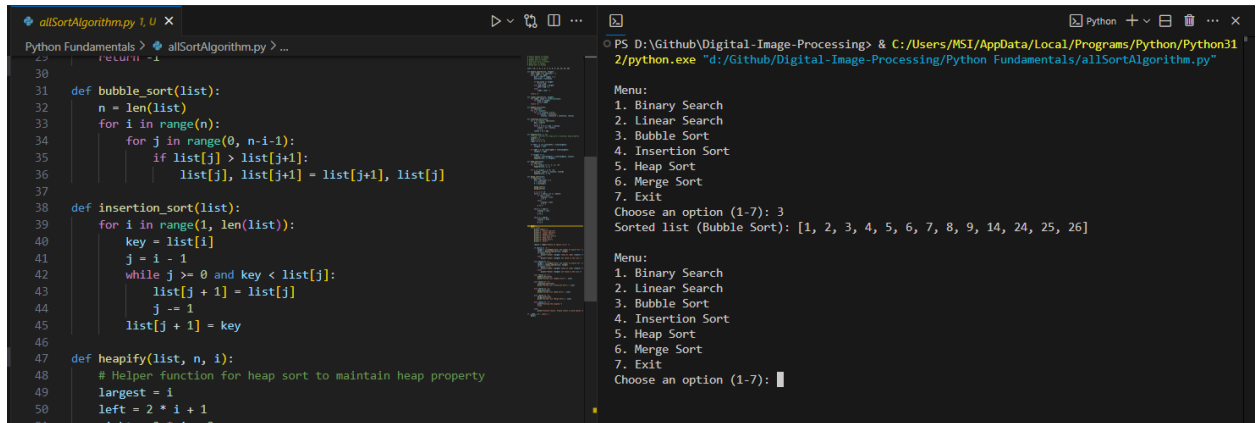
The screenshot shows a Python IDE with a file named `allSortAlgorithm.py`. The code implements a linear search function and a menu-driven interface. The list being searched is `[5, 3, 8, 1, 2, 7, 4, 6, 9, 14, 24, 25, 26]`. The linear search function is defined as follows:

```
def linear_search(list, target):  
    for index, value in enumerate(list):  
        if value == target:  
            return index  
    return -1
```

The execution output shows the menu being displayed, the user selecting option 2 (Linear Search), entering the number 5 to search for, and the program returning the index 0.

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/App2/python.exe "d:/Github/Digital-Image-Processing/Python F  
Menu:  
1. Binary Search  
2. Linear Search  
3. Bubble Sort  
4. Insertion Sort  
5. Heap Sort  
6. Merge Sort  
7. Exit  
Choose an option (1-7): 2  
Enter the number to search for: 5  
Number 5 found at index 0.  
  
Menu:  
1. Binary Search  
2. Linear Search  
3. Bubble Sort  
4. Insertion Sort  
5. Heap Sort  
6. Merge Sort  
7. Exit  
Choose an option (1-7):
```

## 21. Bubble Sort in Python



The screenshot shows a Python IDE with a file named `allSortAlgorithm.py`. The code implements three sorting algorithms: `bubble_sort`, `insertion_sort`, and `heapify`. The `bubble_sort` function is defined as follows:

```
def bubble_sort(list):
    n = len(list)
    for i in range(n):
        for j in range(0, n-i-1):
            if list[j] > list[j+1]:
                list[j], list[j+1] = list[j+1], list[j]
```

The `insertion_sort` function is defined as follows:

```
def insertion_sort(list):
    for i in range(1, len(list)):
        key = list[i]
        j = i - 1
        while j >= 0 and key < list[j]:
            list[j + 1] = list[j]
            j -= 1
        list[j + 1] = key
```

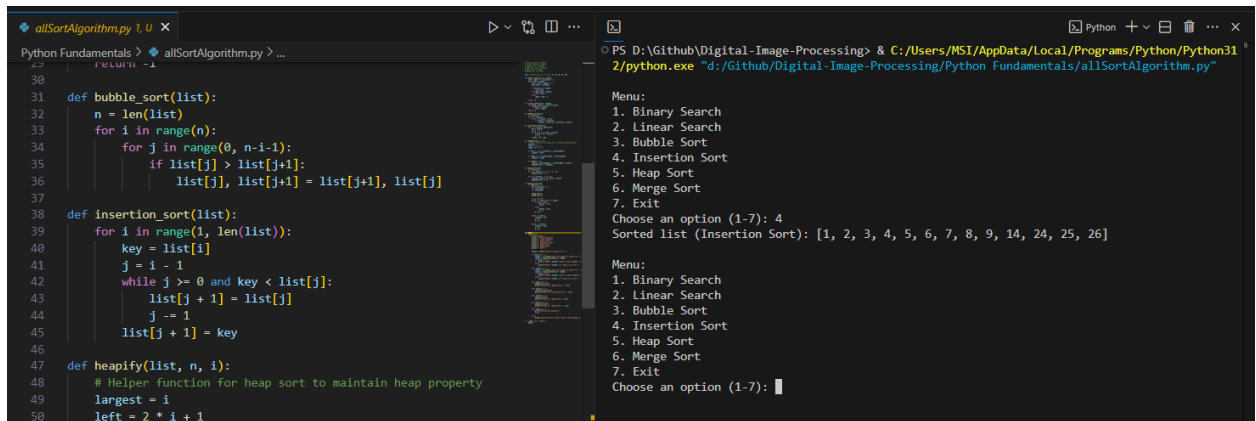
The `heapify` function is defined as follows:

```
def heapify(list, n, i):
    # Helper function for heap sort to maintain heap property
    largest = i
    left = 2 * i + 1
    right = 2 * i + 2
```

The execution output shows the following menu and sorted list:

```
Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7): 3
Sorted list (Bubble Sort): [1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 24, 25, 26]
```

## 22. Insertion Sort in Python



The screenshot shows the same Python IDE with the `allSortAlgorithm.py` file. The `insertion_sort` function is defined as follows:

```
def insertion_sort(list):
    for i in range(1, len(list)):
        key = list[i]
        j = i - 1
        while j >= 0 and key < list[j]:
            list[j + 1] = list[j]
            j -= 1
        list[j + 1] = key
```

The `heapify` function is defined as follows:

```
def heapify(list, n, i):
    # Helper function for heap sort to maintain heap property
    largest = i
    left = 2 * i + 1
    right = 2 * i + 2
```

The execution output shows the following menu and sorted list:

```
Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7): 4
Sorted list (Insertion Sort): [1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 24, 25, 26]
```

## 23. Heap Sort in Python

```
allSortAlgorithm.py 1, U X
Python Fundamentals > allSortAlgorithm.py > ...
45 list[j + 1] = key
46
47 def heapify(list, n, i):
48     # Helper function for heap sort to maintain heap property
49     largest = i
50     left = 2 * i + 1
51     right = 2 * i + 2
52
53     if left < n and list[left] > list[largest]:
54         largest = left
55
56     if right < n and list[right] > list[largest]:
57         largest = right
58
59     if largest != i:
60         list[i], list[largest] = list[largest], list[i]
61         heapify(list, n, largest)
62
63 def heap_sort(list):
64     n = len(list)
65     for i in range(n // 2 - 1, -1, -1):
66         heapify(list, n, i)
67
68     for i in range(n - 1, 0, -1):
69         list[0], list[i] = list[i], list[0]
70         heapify(list, i, 0)
71
72 def merge_sort(list):
73     if len(list) > 1:
```

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/allSortAlgorithm.py"
Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7): 5
Sorted list (Heap Sort): [1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 24, 25, 26]

Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7):
```

## 24. Merge Sort in Python

```
allSortAlgorithm.py 1, U X
Python Fundamentals > allSortAlgorithm.py > ...
71
72 def merge_sort(list):
73     if len(list) > 1:
74         mid = len(list) // 2
75         L = list[:mid]
76         R = list[mid:]
77
78         merge_sort(L)
79         merge_sort(R)
80
81         i = j = k = 0
82         while i < len(L) and j < len(R):
83             if L[i] < R[j]:
84                 list[k] = L[i]
85                 i += 1
86             else:
87                 list[k] = R[j]
88                 j += 1
89             k += 1
90
91         while i < len(L):
92             list[k] = L[i]
93             i += 1
94             k += 1
95
96         while j < len(R):
97             list[k] = R[j]
98             j += 1
99             k += 1
100
101 def main():
102     while True:
103         print("\nMenu:")
104         print("1. Binary Search")
105         print("2. Linear Search")
106         print("3. Bubble Sort")
107         print("4. Insertion Sort")
108         print("5. Heap Sort")
109         print("6. Merge Sort")
```

```
PS D:\Github\Digital-Image-Processing> & C:/Users/MSI/AppData/Local/Programs/Python/Python312/python.exe "d:/Github/Digital-Image-Processing/Python Fundamentals/allSortAlgorithm.py"
Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7): 6
Sorted list (Merge Sort): [1, 2, 3, 4, 5, 6, 7, 8, 9, 14, 24, 25, 26]

Menu:
1. Binary Search
2. Linear Search
3. Bubble Sort
4. Insertion Sort
5. Heap Sort
6. Merge Sort
7. Exit
Choose an option (1-7):
```