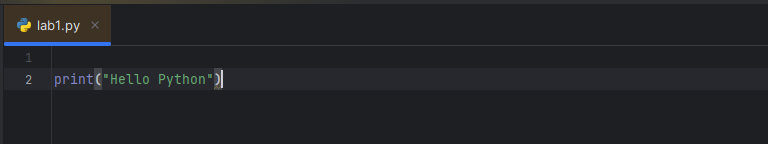
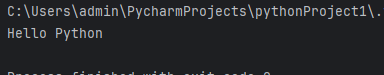
List of Python Programs for Assignment 1 (October 08, ‘24)

Problem 1: Python program to print "Hello Python"

Python code:

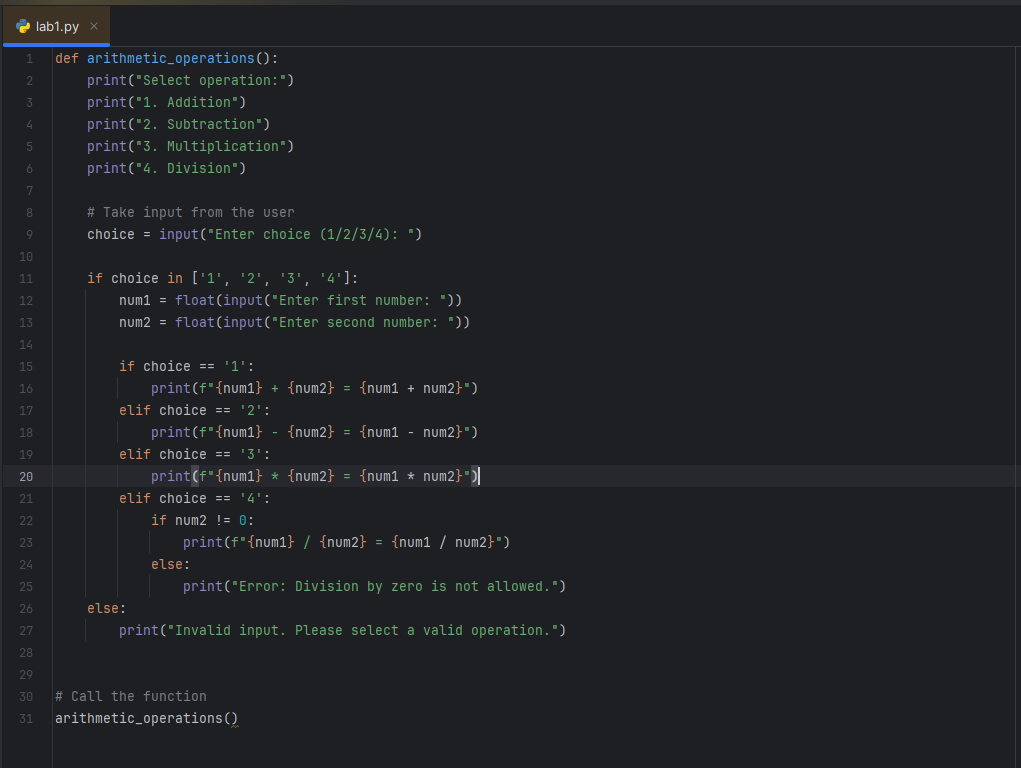
****

Result:

****

Problem 2. Python program to do arithmetical operations

Python code:

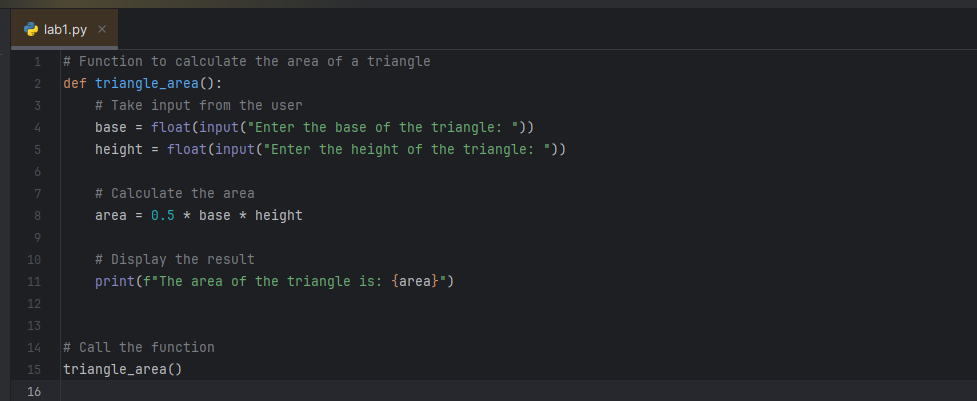


Result:

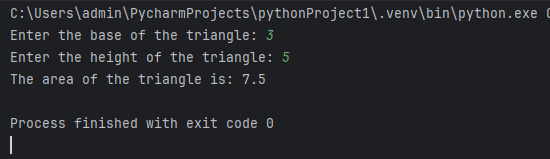


Problem 3. Python program to find the area of a triangle

Python code:

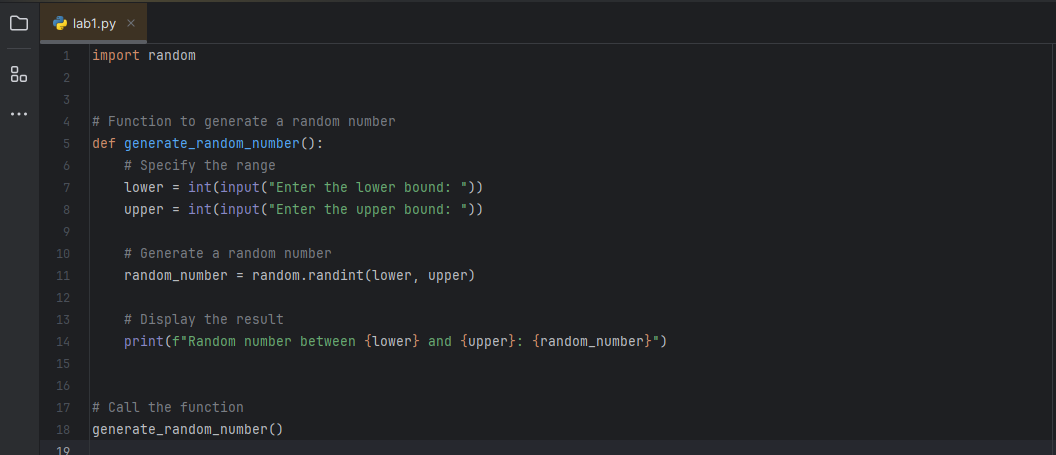


Result:

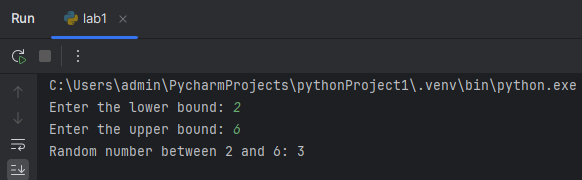


Problem 4. Python program to generate a random number

Python code:

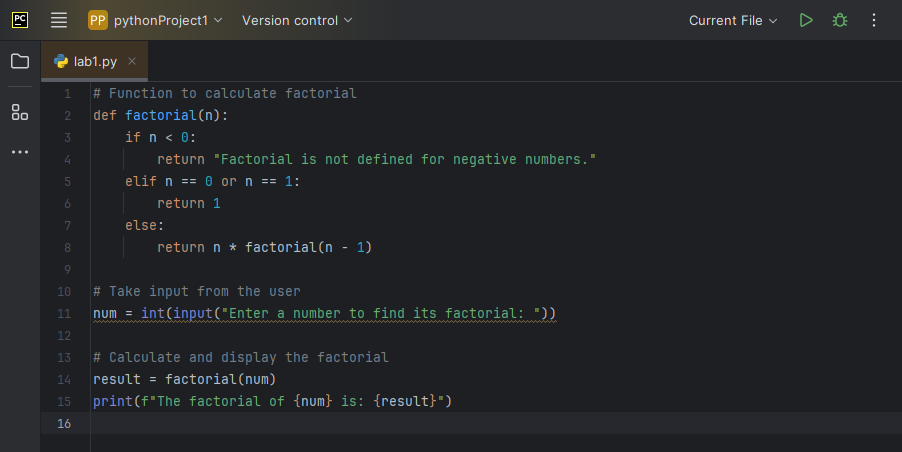


Result:

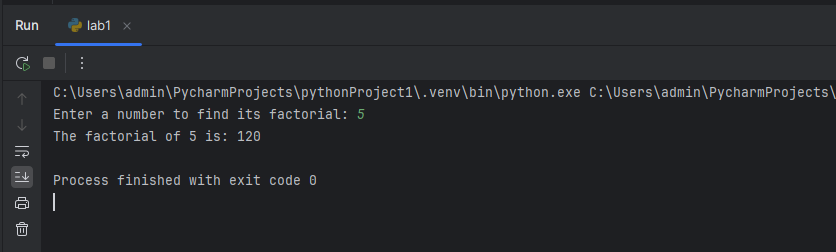


Problem 5. Python Program to Find the Factorial of a Number

Python code:

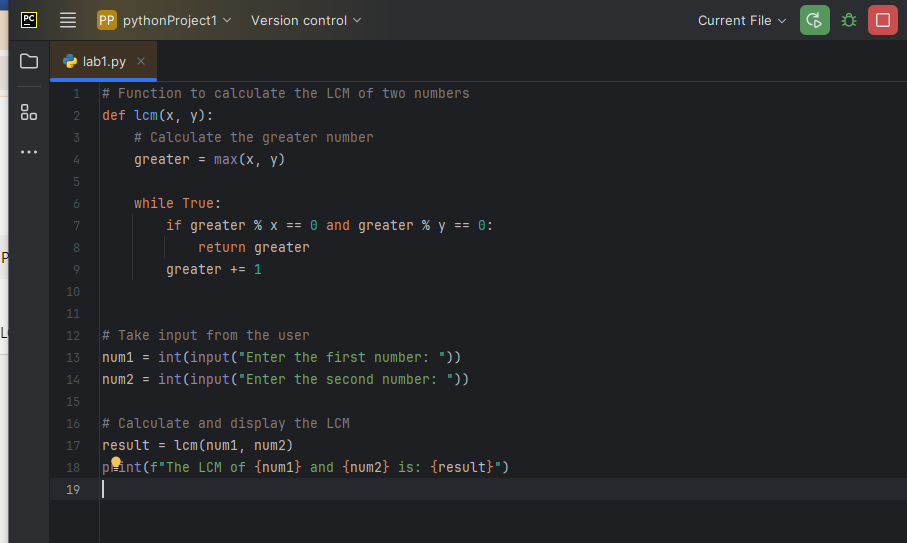
****

Result:

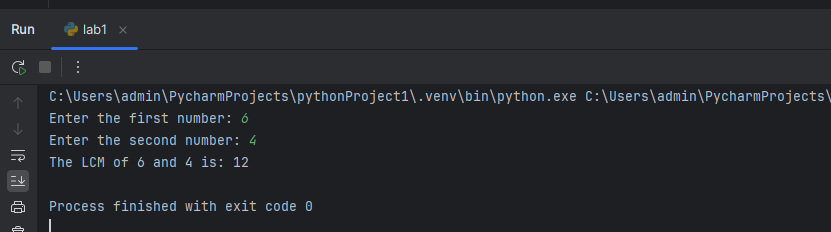
****

Problem 6. Python Program to Find LCM

Python code:

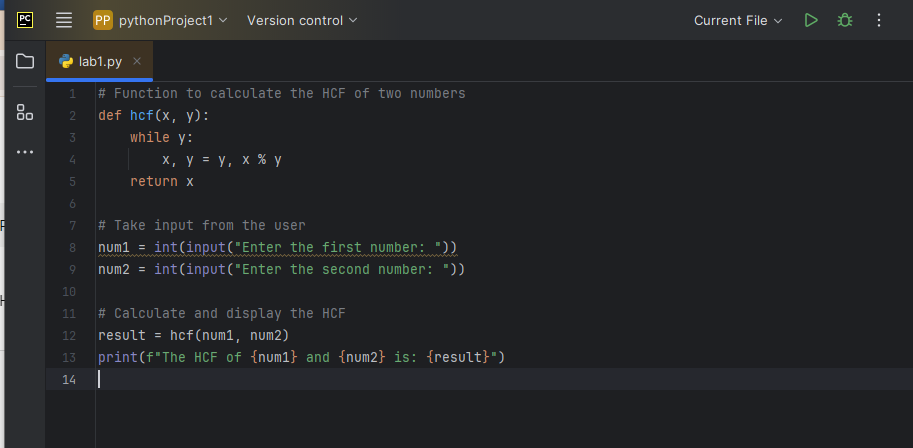
****

Result:

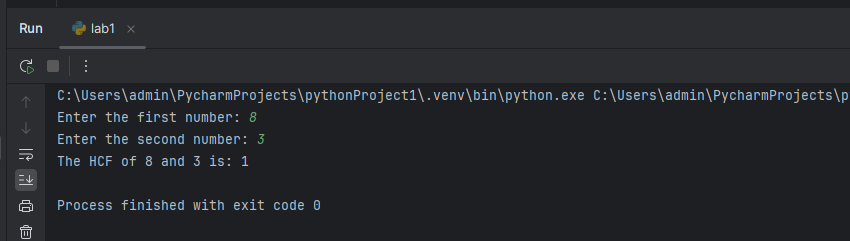
****

Problem 7. Python Program to Find HCF

Python code:

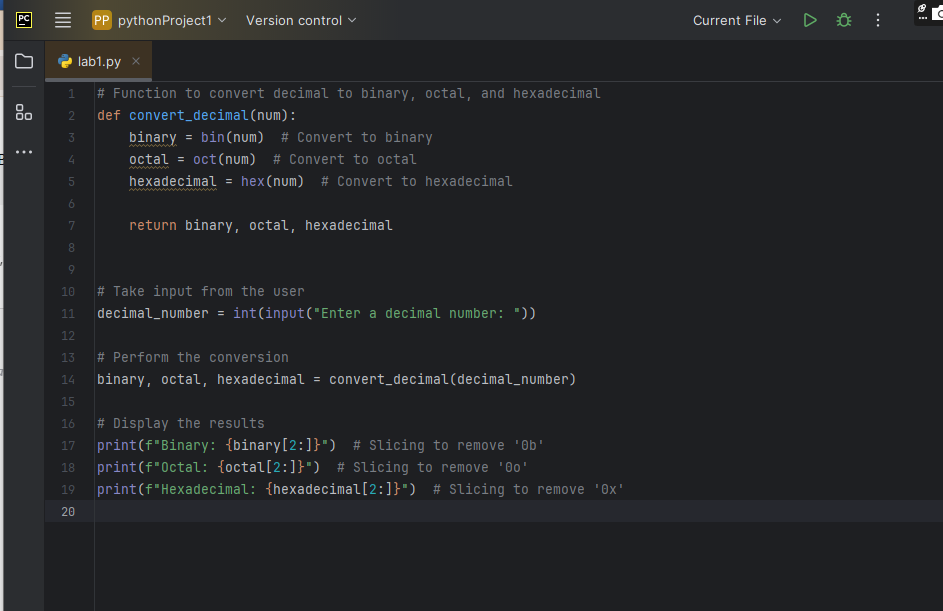
****

Result:

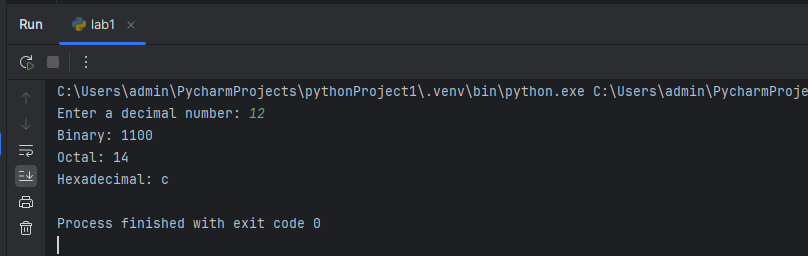
****

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

Python code:

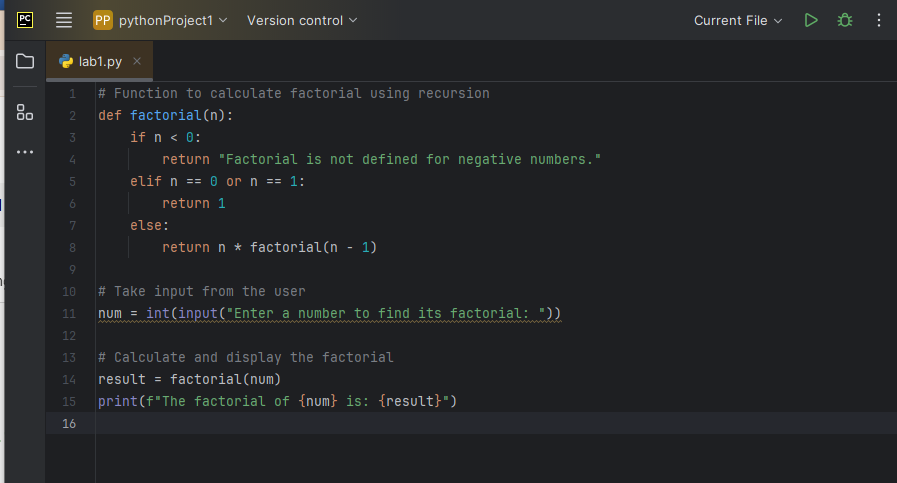


Result:

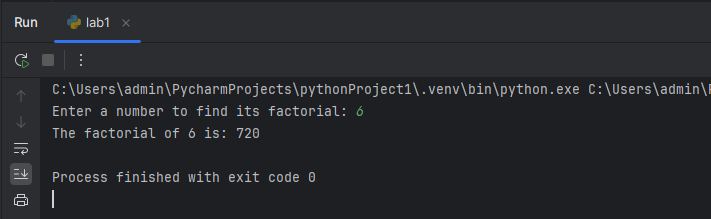


Problem 9. Python Program to Find Factorial of Number Using Recursion

Python code:

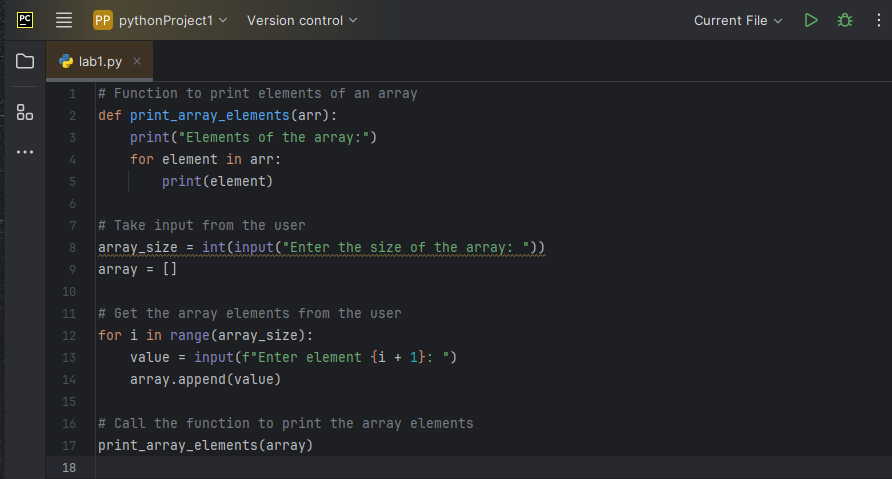
****

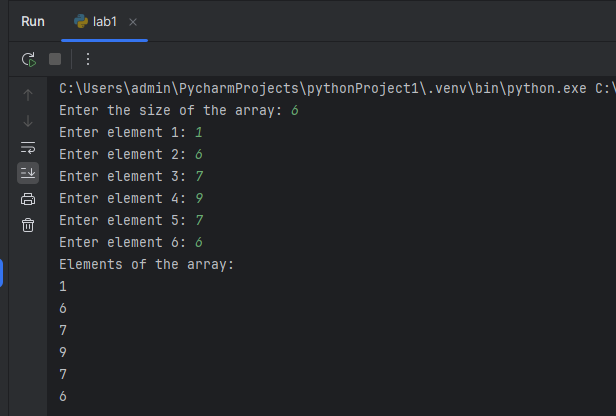
Result:

****

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

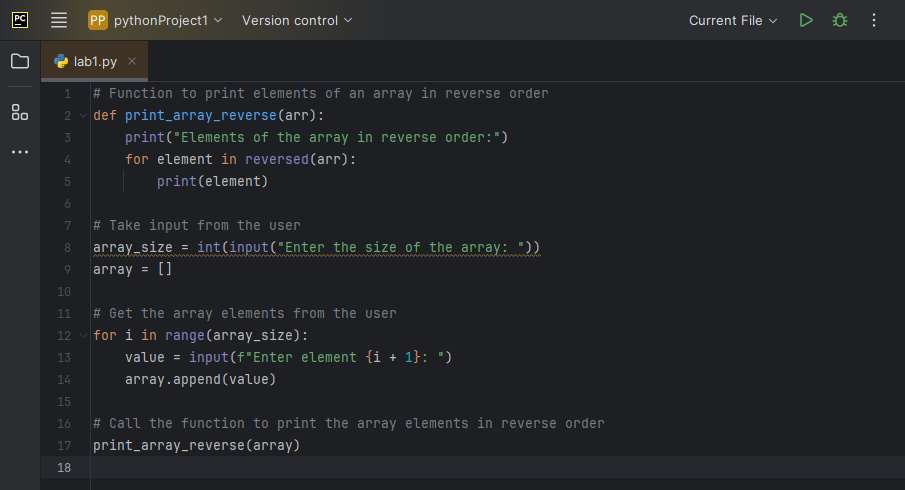
Python code:

****

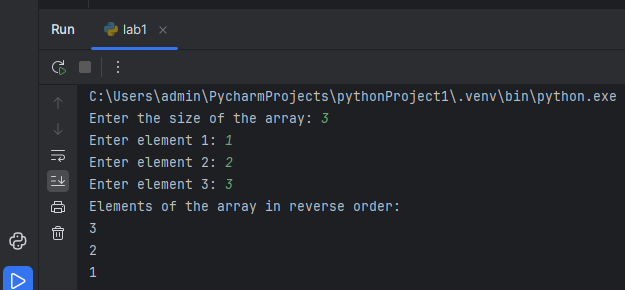
Result: ****

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

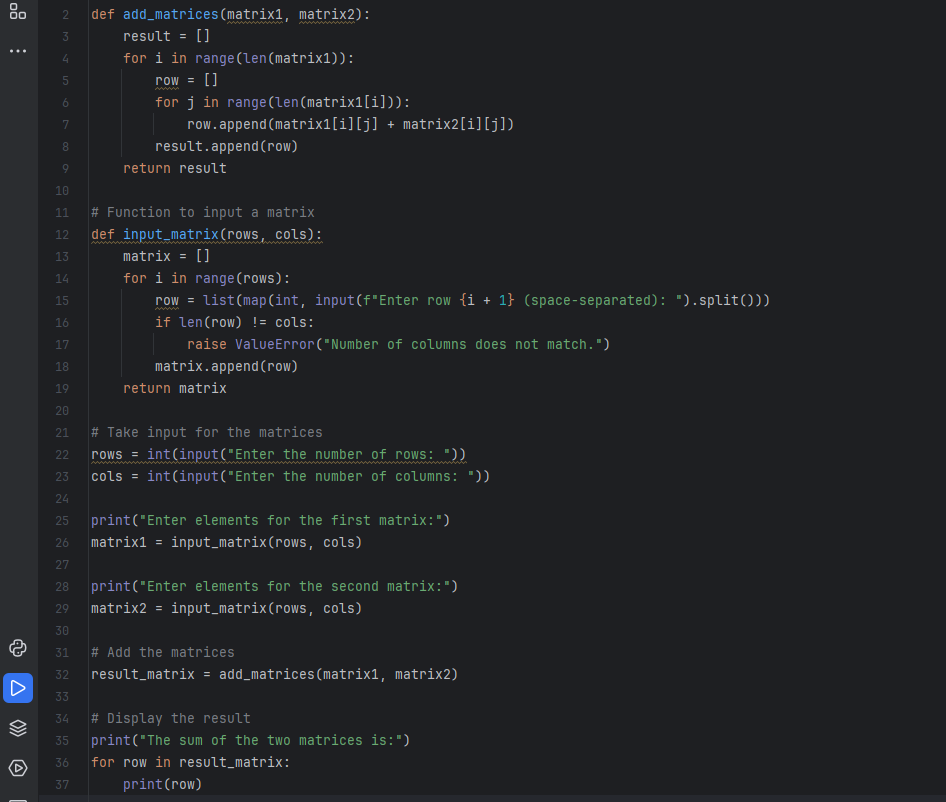
Python code:

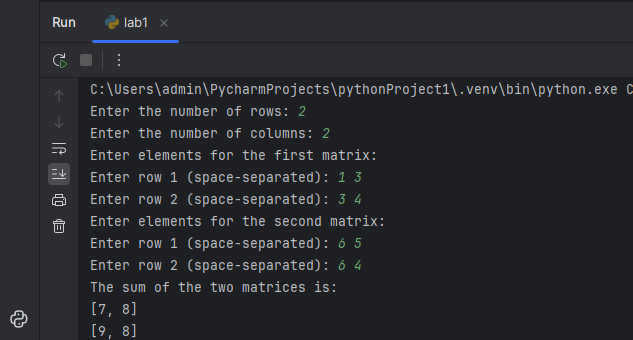
****

Result:

****

Problem 12. Python Program to Add Two Matrices

Python code: ****

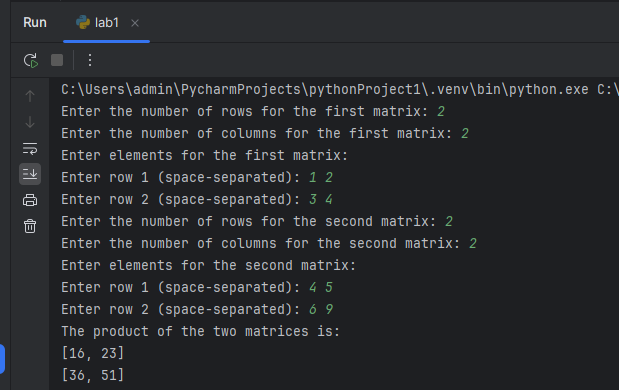
Result: 

Problem 13. Python Program to Multiply Two Matrices

Python code:

def multiply\_matrices(matrix1, matrix2):  
 # Get the dimensions of the matrices  
 rows1 = len(matrix1)  
 cols1 = len(matrix1[0])  
 rows2 = len(matrix2)  
 cols2 = len(matrix2[0])  
  
 # Check if multiplication is possible  
 if cols1 != rows2:  
 raise ValueError(  
 "Number of columns in the first matrix must be equal to the number of rows in the second matrix.")  
  
 # Initialize the result matrix with zeros  
 result = [[0 for \_ in range(cols2)] for \_ in range(rows1)]  
  
 # Perform multiplication  
 for i in range(rows1):  
 for j in range(cols2):  
 for k in range(cols1):  
 result[i][j] += matrix1[i][k] \* matrix2[k][j]  
  
 return result  
  
  
# Function to input a matrix  
def input\_matrix(rows, cols):  
 matrix = []  
 for i in range(rows):  
 row = list(map(int, input(f"Enter row {i + 1} (space-separated): ").split()))  
 if len(row) != cols:  
 raise ValueError("Number of columns does not match.")  
 matrix.append(row)  
 return matrix  
  
  
# Take input for the matrices  
rows1 = int(input("Enter the number of rows for the first matrix: "))  
cols1 = int(input("Enter the number of columns for the first matrix: "))  
  
print("Enter elements for the first matrix:")  
matrix1 = input\_matrix(rows1, cols1)  
  
rows2 = int(input("Enter the number of rows for the second matrix: "))  
cols2 = int(input("Enter the number of columns for the second matrix: "))  
  
print("Enter elements for the second matrix:")  
matrix2 = input\_matrix(rows2, cols2)  
  
# Multiply the matrices  
result\_matrix = multiply\_matrices(matrix1, matrix2)  
  
# Display the result  
print("The product of the two matrices is:")  
for row in result\_matrix:  
 print(row)

Result:

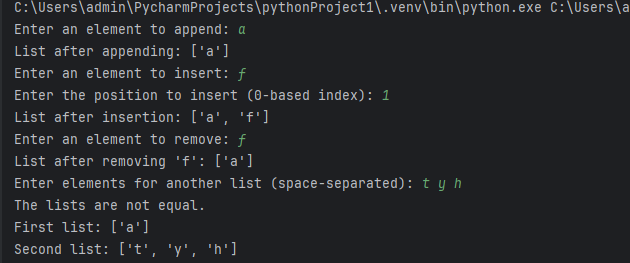


Problem 14. Python Program to append element in the list and update list with insertion of elements, removing an element, comparison of two lists, etc.

Python code:

# Function to demonstrate list operations  
def list\_operations():  
 # Initialize an empty list  
 my\_list = []  
  
 # Append an element  
 element\_to\_append = input("Enter an element to append: ")  
 my\_list.append(element\_to\_append)  
 print(f"List after appending: {my\_list}")  
  
 # Insert an element at a specific position  
 element\_to\_insert = input("Enter an element to insert: ")  
 position\_to\_insert = int(input("Enter the position to insert (0-based index): "))  
 my\_list.insert(position\_to\_insert, element\_to\_insert)  
 print(f"List after insertion: {my\_list}")  
  
 # Remove an element  
 element\_to\_remove = input("Enter an element to remove: ")  
 if element\_to\_remove in my\_list:  
 my\_list.remove(element\_to\_remove)  
 print(f"List after removing '{element\_to\_remove}': {my\_list}")  
 else:  
 print(f"Element '{element\_to\_remove}' not found in the list.")  
  
 # Compare with another list  
 another\_list = input("Enter elements for another list (space-separated): ").split()  
 if my\_list == another\_list:  
 print("Both lists are equal.")  
 else:  
 print("The lists are not equal.")  
  
 print(f"First list: {my\_list}")  
 print(f"Second list: {another\_list}")  
  
  
# Call the function to perform list operations  
list\_operations()

Result:

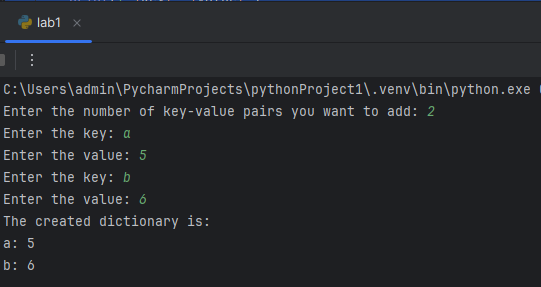
****

Problem 15. Python Program to create a dictionary:

Python code:

# Function to create a dictionary  
def create\_dictionary():  
 my\_dict = {}  
  
 # Number of entries  
 num\_entries = int(input("Enter the number of key-value pairs you want to add: "))  
  
 for \_ in range(num\_entries):  
 key = input("Enter the key: ")  
 value = input("Enter the value: ")  
 my\_dict[key] = value  
  
 return my\_dict  
  
  
# Function to display the dictionary  
def display\_dictionary(my\_dict):  
 print("The created dictionary is:")  
 for key, value in my\_dict.items():  
 print(f"{key}: {value}")  
  
  
# Main execution  
my\_dict = create\_dictionary()  
display\_dictionary(my\_dict)

Result:

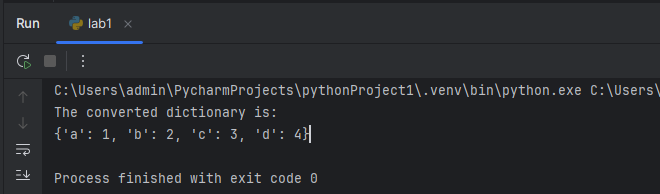
****

Problem 16. Python Program to convert list to dictionary

Python code:

# Function to convert a list of tuples to a dictionary  
def list\_to\_dictionary(tuples\_list):  
 return dict(tuples\_list)  
  
# Main execution  
# Example list of tuples  
tuples\_list = [('a', 1), ('b', 2), ('c', 3), ('d', 4)]  
  
# Convert list to dictionary  
result\_dict = list\_to\_dictionary(tuples\_list)  
  
# Display the result  
print("The converted dictionary is:")  
print(result\_dict)

Result:

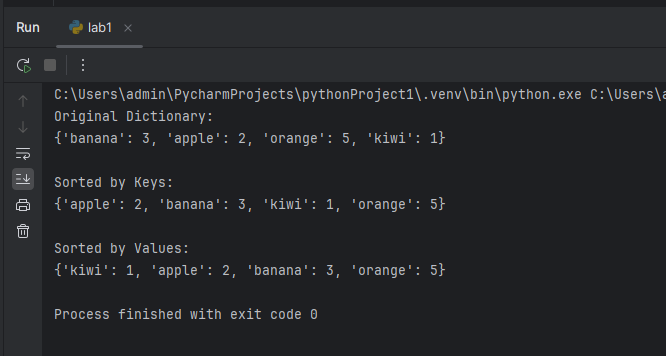
****

Problem 17. Python Program to sort a dictionary

Python code:

# Function to sort a dictionary  
def sort\_dictionary(input\_dict):  
 # Sort by keys  
 sorted\_by\_keys = dict(sorted(input\_dict.items()))  
  
 # Sort by values  
 sorted\_by\_values = dict(sorted(input\_dict.items(), key=lambda item: item[1]))  
  
 return sorted\_by\_keys, sorted\_by\_values  
  
  
# Main execution  
# Example dictionary  
my\_dict = {  
 'banana': 3,  
 'apple': 2,  
 'orange': 5,  
 'kiwi': 1  
}  
  
# Sort the dictionary  
sorted\_keys, sorted\_values = sort\_dictionary(my\_dict)  
  
# Display the results  
print("Original Dictionary:")  
print(my\_dict)  
print("\nSorted by Keys:")  
print(sorted\_keys)  
print("\nSorted by Values:")  
print(sorted\_values)

Result:

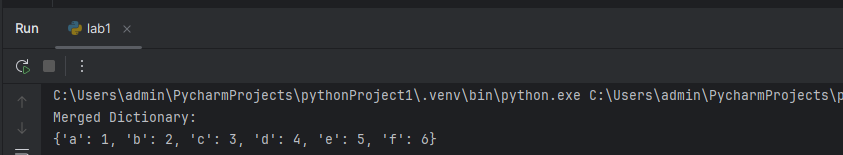
****

Problem 18. Python Program to Merge two Dictionaries

Python code:

# Function to merge two dictionaries  
def merge\_dictionaries(dict1, dict2):  
 # Merge using the update() method  
 merged\_dict = dict1.copy() # Create a copy to avoid modifying the original  
 merged\_dict.update(dict2)  
 return merged\_dict  
  
# Main execution  
# Example dictionaries  
dict1 = {  
 'a': 1,  
 'b': 2,  
 'c': 3  
}  
  
dict2 = {  
 'd': 4,  
 'e': 5,  
 'f': 6  
}  
  
# Merge the dictionaries  
result\_dict = merge\_dictionaries(dict1, dict2)  
  
# Display the result  
print("Merged Dictionary:")  
print(result\_dict)

Result:

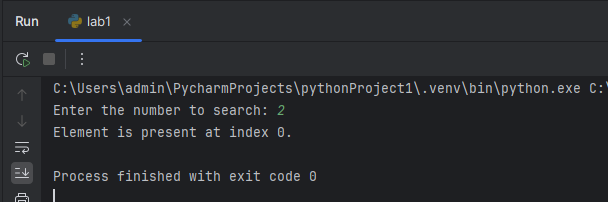
****

Problem 19. Binary Search in Python

Python code:

# Function to perform binary search  
def binary\_search(arr, target):  
 left, right = 0, len(arr) - 1  
  
 while left <= right:  
 mid = left + (right - left) // 2 # Calculate mid index  
  
 # Check if the target is present at mid  
 if arr[mid] == target:  
 return mid  
 # If the target is greater, ignore the left half  
 elif arr[mid] < target:  
 left = mid + 1  
 # If the target is smaller, ignore the right half  
 else:  
 right = mid - 1  
  
 return -1 # Target not found  
  
  
# Main execution  
# Example sorted array  
arr = [2, 3, 4, 10, 40]  
target = int(input("Enter the number to search: "))  
  
# Perform binary search  
result = binary\_search(arr, target)  
  
# Display the result  
if result != -1:  
 print(f"Element is present at index {result}.")  
else:  
 print("Element is not present in the array.")

Result:

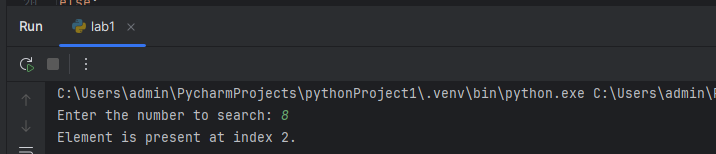
****

Problem 20. Linear Search in Python

Python code:

# Function to perform linear search  
def linear\_search(arr, target):  
 for index, element in enumerate(arr):  
 # Check if the target is present at the current index  
 if element == target:  
 return index  
 return -1 # Target not found  
  
# Main execution  
# Example array  
arr = [5, 3, 8, 6, 2, 7]  
target = int(input("Enter the number to search: "))  
  
# Perform linear search  
result = linear\_search(arr, target)  
  
# Display the result  
if result != -1:  
 print(f"Element is present at index {result}.")  
else:  
 print("Element is not present in the array.")

Result:

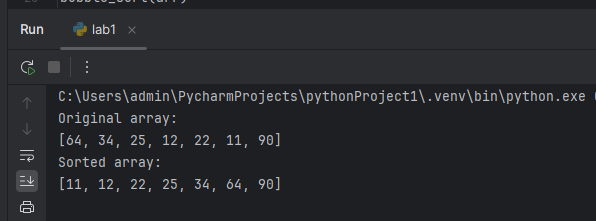
****

Problem 21. Bubble Sort in Python

Python code:

# Function to perform bubble sort  
def bubble\_sort(arr):  
 n = len(arr)  
 # Traverse through all array elements  
 for i in range(n):  
 # Last i elements are already sorted  
 for j in range(0, n - i - 1):  
 # Swap if the element found is greater than the next element  
 if arr[j] > arr[j + 1]:  
 arr[j], arr[j + 1] = arr[j + 1], arr[j]  
  
# Main execution  
# Example array  
arr = [64, 34, 25, 12, 22, 11, 90]  
  
print("Original array:")  
print(arr)  
  
# Perform bubble sort  
bubble\_sort(arr)  
  
# Display the sorted array  
print("Sorted array:")  
print(arr)

Result:

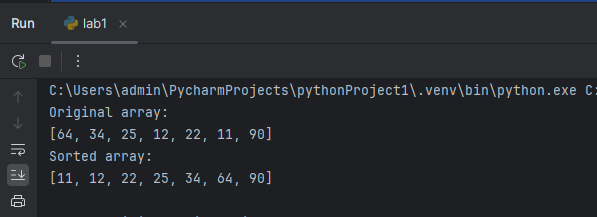
****

Problem 22. Insertion Sort in Python

**Python code:**

# Function to perform bubble sort  
def bubble\_sort(arr):  
 n = len(arr)  
 # Traverse through all array elements  
 for i in range(n):  
 # Last i elements are already sorted  
 for j in range(0, n - i - 1):  
 # Swap if the element found is greater than the next element  
 if arr[j] > arr[j + 1]:  
 arr[j], arr[j + 1] = arr[j + 1], arr[j]  
  
# Main execution  
# Example array  
arr = [64, 34, 25, 12, 22, 11, 90]  
  
print("Original array:")  
print(arr)  
  
# Perform bubble sort  
bubble\_sort(arr)  
  
# Display the sorted array  
print("Sorted array:")  
print(arr)

**Result:**

****