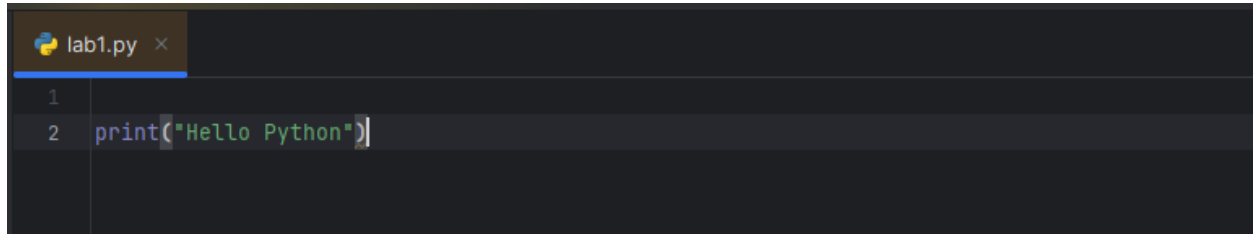


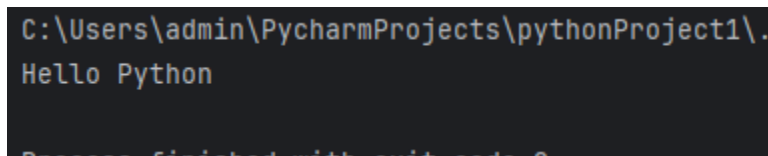
List of Python Programs for Assignment 1 (October 08, '24)

Problem 1: Python program to print "Hello Python"

Python code:

A screenshot of a Python IDE window titled 'lab1.py'. The code editor shows two lines of code: line 1 is empty, and line 2 contains the statement `print("Hello Python")`. The text is color-coded: `print` is blue, `(` is blue, `"Hello Python"` is green, and `)` is blue. The cursor is at the end of the line.

Result:

A screenshot of a terminal window. The first line shows the file path `C:\Users\admin\PycharmProjects\pythonProject1\.`. The second line shows the output `Hello Python`. The third line shows the prompt `Process finished with exit code 0`.

Problem 2. Python program to do arithmetical operations

Python code:

```
lab1.py x
1 def arithmetic_operations():
2     print("Select operation:")
3     print("1. Addition")
4     print("2. Subtraction")
5     print("3. Multiplication")
6     print("4. Division")
7
8     # Take input from the user
9     choice = input("Enter choice (1/2/3/4): ")
10
11     if choice in ['1', '2', '3', '4']:
12         num1 = float(input("Enter first number: "))
13         num2 = float(input("Enter second number: "))
14
15         if choice == '1':
16             print(f"{num1} + {num2} = {num1 + num2}")
17         elif choice == '2':
18             print(f"{num1} - {num2} = {num1 - num2}")
19         elif choice == '3':
20             print(f"{num1} * {num2} = {num1 * num2}")
21         elif choice == '4':
22             if num2 != 0:
23                 print(f"{num1} / {num2} = {num1 / num2}")
24             else:
25                 print("Error: Division by zero is not allowed.")
26         else:
27             print("Invalid input. Please select a valid operation.")
28
29
30 # Call the function
31 arithmetic_operations()
```

Result:

```
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\
Select operation:
1. Addition
2. Subtraction
3. Multiplication
4. Division
Enter choice (1/2/3/4): 2
Enter first number: 6
Enter second number: 5
6.0 - 5.0 = 1.0

Process finished with exit code 0
```

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

Python code:

```
lab1.py x
1 # Function to calculate the area of a triangle
2 def triangle_area():
3     # Take input from the user
4     base = float(input("Enter the base of the triangle: "))
5     height = float(input("Enter the height of the triangle: "))
6
7     # Calculate the area
8     area = 0.5 * base * height
9
10    # Display the result
11    print(f"The area of the triangle is: {area}")
12
13
14 # Call the function
15 triangle_area()
16
```

Result:

```
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C
Enter the base of the triangle: 3
Enter the height of the triangle: 5
The area of the triangle is: 7.5

Process finished with exit code 0
|
```

Problem 4. Python program to generate a random number

Python code:

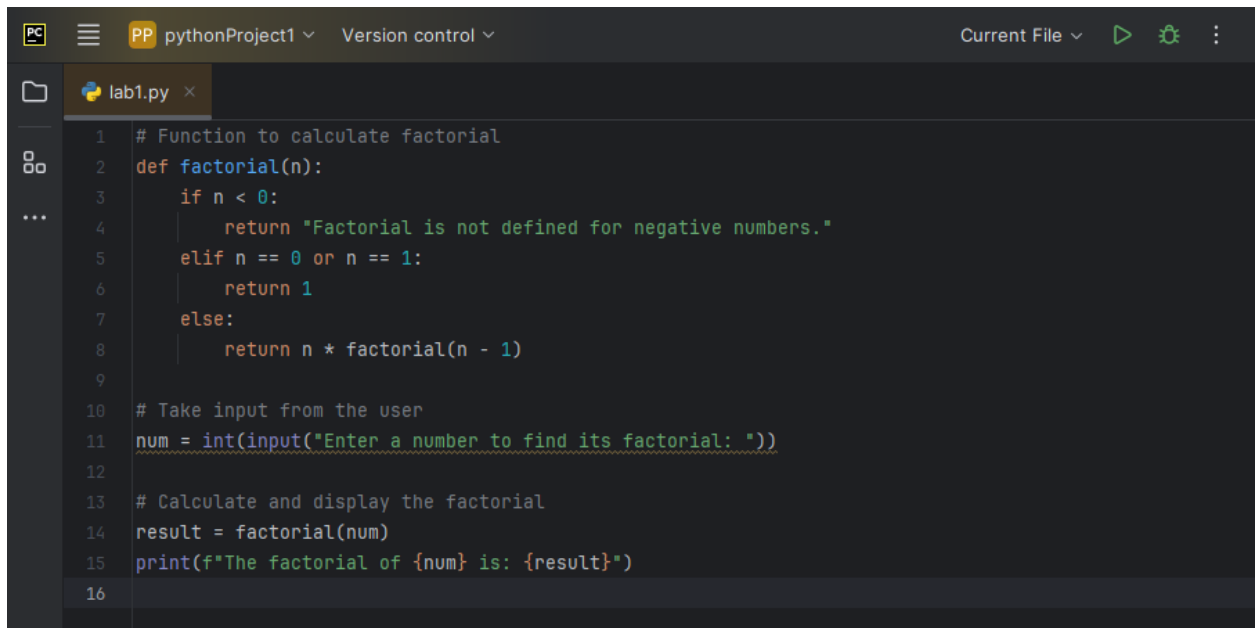
```
lab1.py x
1 import random
2
3
4 # Function to generate a random number
5 def generate_random_number():
6     # Specify the range
7     lower = int(input("Enter the lower bound: "))
8     upper = int(input("Enter the upper bound: "))
9
10    # Generate a random number
11    random_number = random.randint(lower, upper)
12
13    # Display the result
14    print(f"Random number between {lower} and {upper}: {random_number}")
15
16
17 # Call the function
18 generate_random_number()
19
```

Result:

```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe
Enter the lower bound: 2
Enter the upper bound: 6
Random number between 2 and 6: 3
```

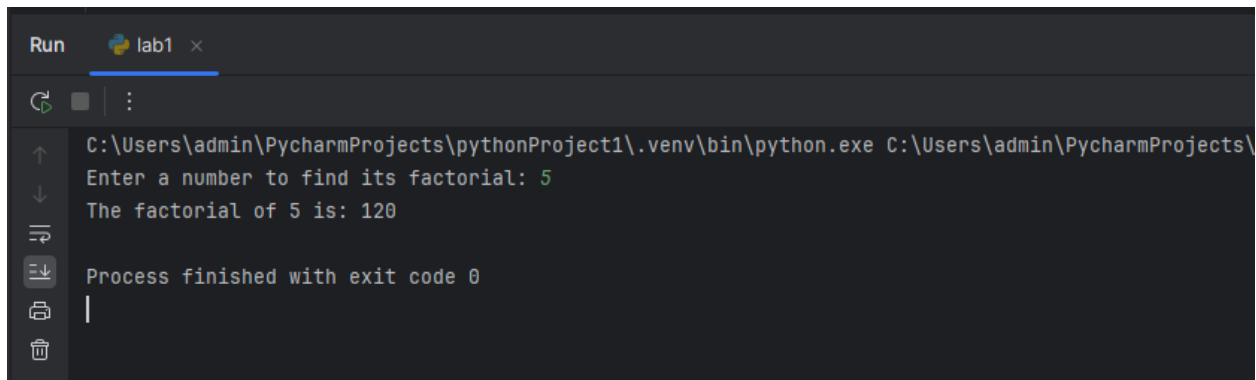
Problem 5. Python Program to Find the Factorial of a Number

Python code:

A screenshot of a Python IDE window titled 'pythonProject1'. The file 'lab1.py' is open. The code defines a recursive factorial function and takes user input to calculate and print the factorial of a number. The code is as follows:

```
1 # Function to calculate factorial
2 def factorial(n):
3     if n < 0:
4         return "Factorial is not defined for negative numbers."
5     elif n == 0 or n == 1:
6         return 1
7     else:
8         return n * factorial(n - 1)
9
10 # Take input from the user
11 num = int(input("Enter a number to find its factorial: "))
12
13 # Calculate and display the factorial
14 result = factorial(num)
15 print(f"The factorial of {num} is: {result}")
16
```

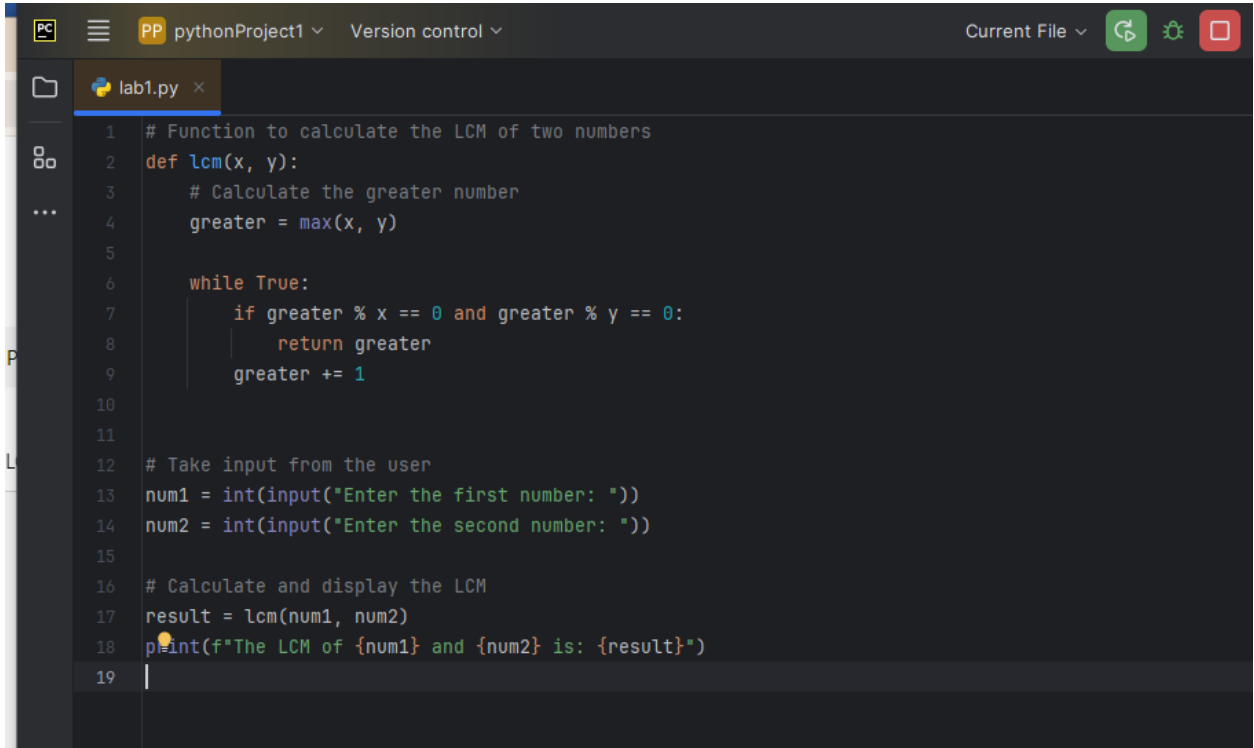
Result:

A screenshot of a terminal window titled 'Run lab1'. It shows the execution of the Python program. The user enters '5' as input, and the program outputs 'The factorial of 5 is: 120'. The process finished with exit code 0.

```
Run lab1
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProjects\
Enter a number to find its factorial: 5
The factorial of 5 is: 120
Process finished with exit code 0
```

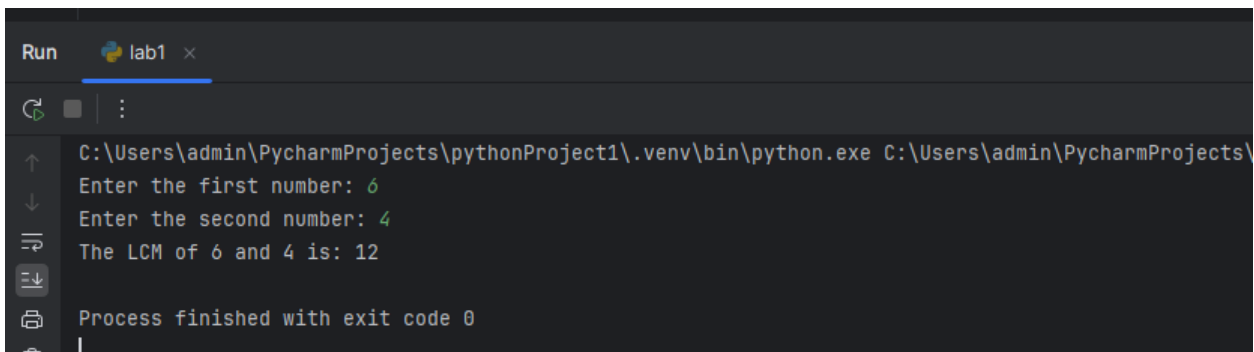
Problem 6. Python Program to Find LCM

Python code:



```
1 # Function to calculate the LCM of two numbers
2 def lcm(x, y):
3     # Calculate the greater number
4     greater = max(x, y)
5
6     while True:
7         if greater % x == 0 and greater % y == 0:
8             return greater
9         greater += 1
10
11
12 # Take input from the user
13 num1 = int(input("Enter the first number: "))
14 num2 = int(input("Enter the second number: "))
15
16 # Calculate and display the LCM
17 result = lcm(num1, num2)
18 print(f"The LCM of {num1} and {num2} is: {result}")
19
```

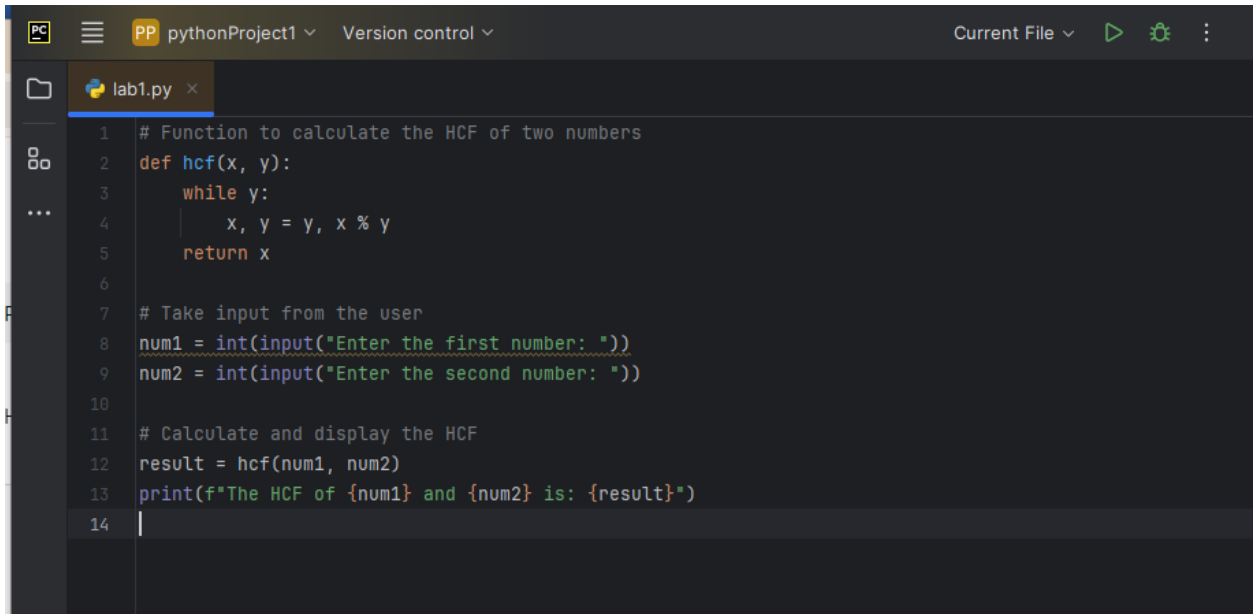
Result:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProjects\
Enter the first number: 6
Enter the second number: 4
The LCM of 6 and 4 is: 12
Process finished with exit code 0
```

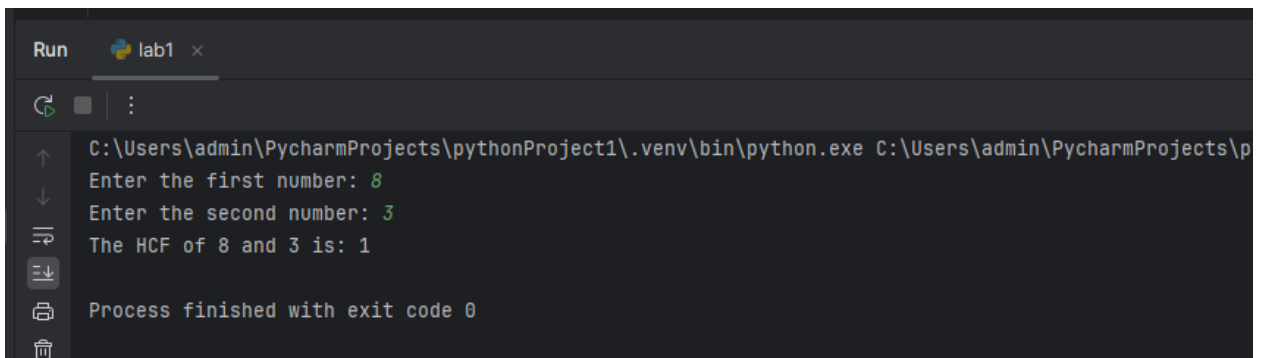
Problem 7. Python Program to Find HCF

Python code:



```
1 # Function to calculate the HCF of two numbers
2 def hcf(x, y):
3     while y:
4         x, y = y, x % y
5     return x
6
7 # Take input from the user
8 num1 = int(input("Enter the first number: "))
9 num2 = int(input("Enter the second number: "))
10
11 # Calculate and display the HCF
12 result = hcf(num1, num2)
13 print(f"The HCF of {num1} and {num2} is: {result}")
14
```

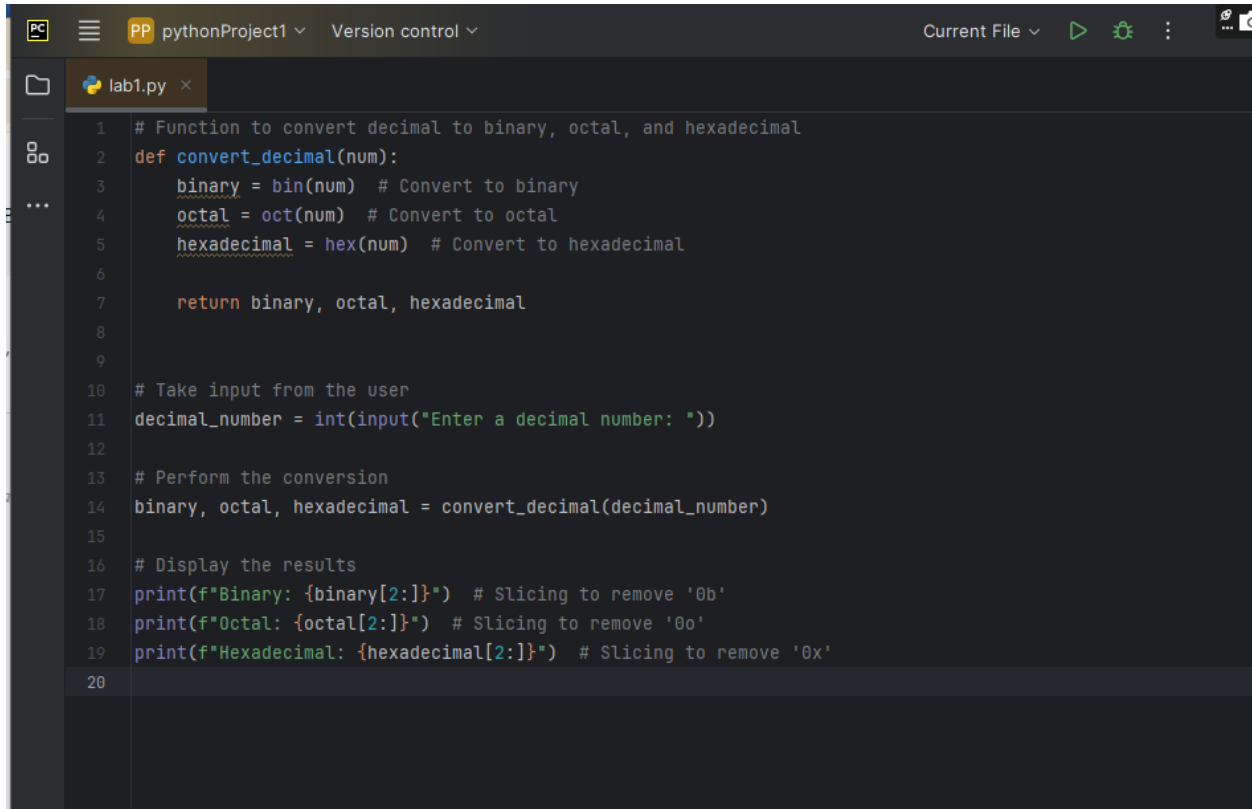
Result:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProjects\p
Enter the first number: 8
Enter the second number: 3
The HCF of 8 and 3 is: 1
Process finished with exit code 0
```

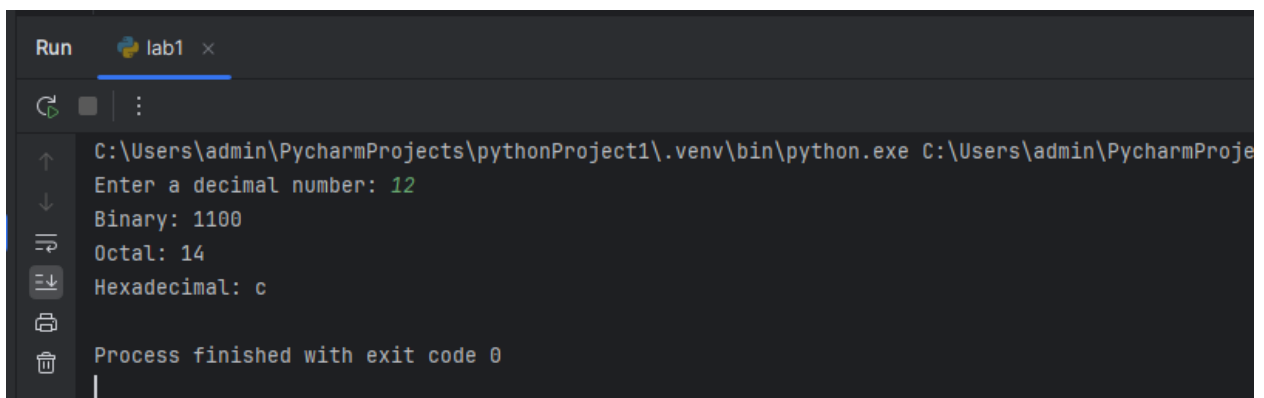
Problem 8. Python Program to Convert Decimal to Binary, Octal and Hexadecimal

Python code:



```
1 # Function to convert decimal to binary, octal, and hexadecimal
2 def convert_decimal(num):
3     binary = bin(num) # Convert to binary
4     octal = oct(num) # Convert to octal
5     hexadecimal = hex(num) # Convert to hexadecimal
6
7     return binary, octal, hexadecimal
8
9
10 # Take input from the user
11 decimal_number = int(input("Enter a decimal number: "))
12
13 # Perform the conversion
14 binary, octal, hexadecimal = convert_decimal(decimal_number)
15
16 # Display the results
17 print(f"Binary: {binary[2:]}") # Slicing to remove '0b'
18 print(f"Octal: {octal[2:]}") # Slicing to remove '0o'
19 print(f"Hexadecimal: {hexadecimal[2:]}") # Slicing to remove '0x'
20
```

Result:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProje
Enter a decimal number: 12
Binary: 1100
Octal: 14
Hexadecimal: c
Process finished with exit code 0
```


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

Python code:

```
1 # Function to calculate factorial using recursion
2 def factorial(n):
3     if n < 0:
4         return "Factorial is not defined for negative numbers."
5     elif n == 0 or n == 1:
6         return 1
7     else:
8         return n * factorial(n - 1)
9
10 # Take input from the user
11 num = int(input("Enter a number to find its factorial: "))
12
13 # Calculate and display the factorial
14 result = factorial(num)
15 print(f"The factorial of {num} is: {result}")
16
```

Result:

```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\
Enter a number to find its factorial: 6
The factorial of 6 is: 720
Process finished with exit code 0
```

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

Python code:

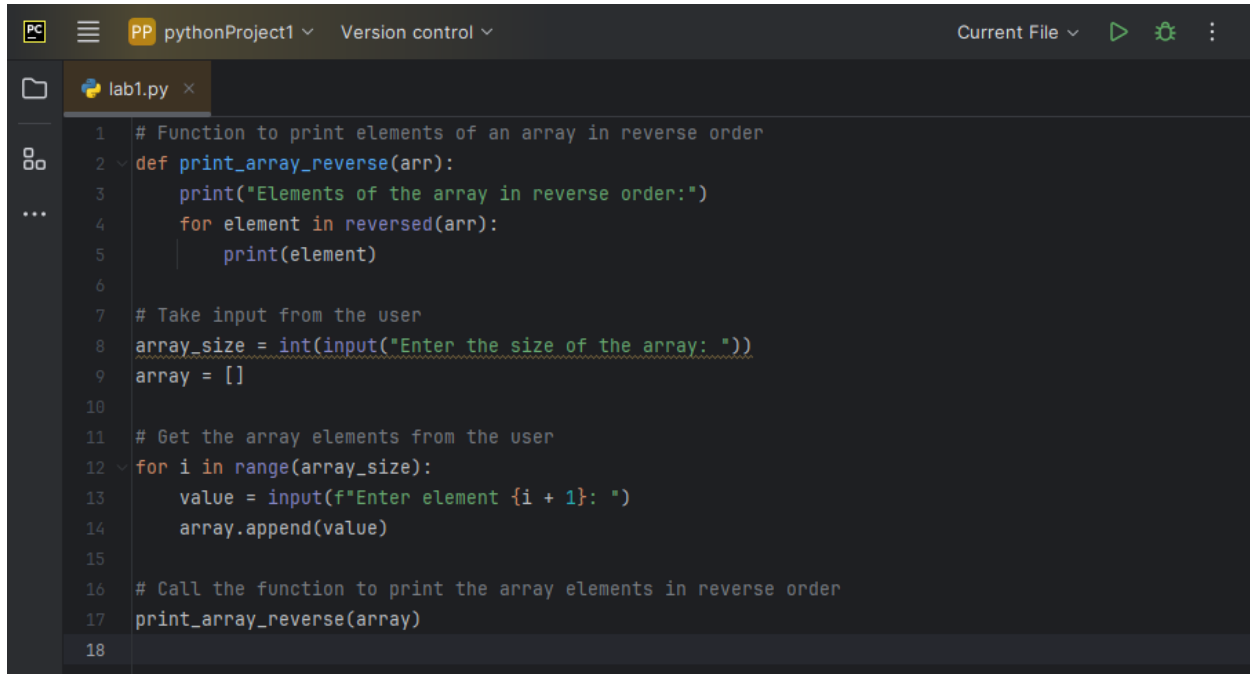
```
pythonProject1 Version control Current File
lab1.py
1 # Function to print elements of an array
2 def print_array_elements(arr):
3     print("Elements of the array:")
4     for element in arr:
5         print(element)
6
7 # Take input from the user
8 array_size = int(input("Enter the size of the array: "))
9 array = []
10
11 # Get the array elements from the user
12 for i in range(array_size):
13     value = input(f"Enter element {i + 1}: ")
14     array.append(value)
15
16 # Call the function to print the array elements
17 print_array_elements(array)
18
```

Result:

```
Run lab1
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\
Enter the size of the array: 6
Enter element 1: 1
Enter element 2: 6
Enter element 3: 7
Enter element 4: 9
Enter element 5: 7
Enter element 6: 6
Elements of the array:
1
6
7
9
7
6
```

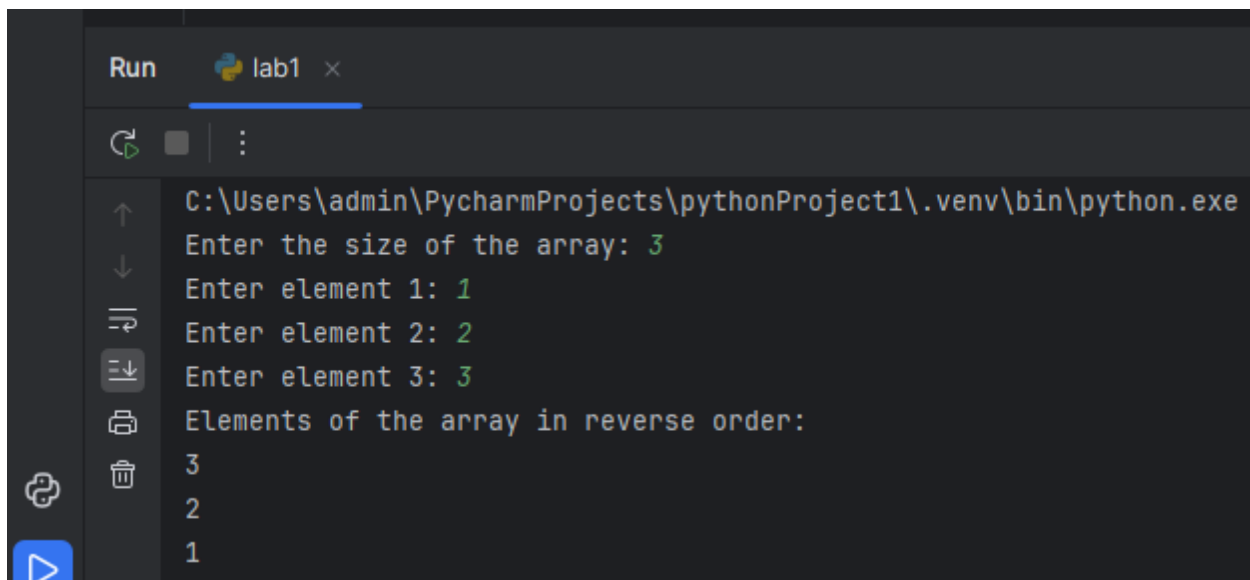
Problem 11. Python program to print the elements of an array in reverse order

Python code:



```
1 # Function to print elements of an array in reverse order
2 def print_array_reverse(arr):
3     print("Elements of the array in reverse order:")
4     for element in reversed(arr):
5         print(element)
6
7 # Take input from the user
8 array_size = int(input("Enter the size of the array: "))
9 array = []
10
11 # Get the array elements from the user
12 for i in range(array_size):
13     value = input(f"Enter element {i + 1}: ")
14     array.append(value)
15
16 # Call the function to print the array elements in reverse order
17 print_array_reverse(array)
18
```

Result:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe
Enter the size of the array: 3
Enter element 1: 1
Enter element 2: 2
Enter element 3: 3
Elements of the array in reverse order:
3
2
1
```

Problem 12. Python Program to Add Two Matrices

Python code:

```
2  def add_matrices(matrix1, matrix2):
3      result = []
4      for i in range(len(matrix1)):
5          row = []
6          for j in range(len(matrix1[i])):
7              row.append(matrix1[i][j] + matrix2[i][j])
8          result.append(row)
9      return result
10
11 # Function to input a matrix
12 def input_matrix(rows, cols):
13     matrix = []
14     for i in range(rows):
15         row = list(map(int, input(f"Enter row {i + 1} (space-separated): ").split()))
16         if len(row) != cols:
17             raise ValueError("Number of columns does not match.")
18         matrix.append(row)
19     return matrix
20
21 # Take input for the matrices
22 rows = int(input("Enter the number of rows: "))
23 cols = int(input("Enter the number of columns: "))
24
25 print("Enter elements for the first matrix:")
26 matrix1 = input_matrix(rows, cols)
27
28 print("Enter elements for the second matrix:")
29 matrix2 = input_matrix(rows, cols)
30
31 # Add the matrices
32 result_matrix = add_matrices(matrix1, matrix2)
33
34 # Display the result
35 print("The sum of the two matrices is:")
36 for row in result_matrix:
37     print(row)
```

Result:

```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C
Enter the number of rows: 2
Enter the number of columns: 2
Enter elements for the first matrix:
Enter row 1 (space-separated): 1 3
Enter row 2 (space-separated): 3 4
Enter elements for the second matrix:
Enter row 1 (space-separated): 6 5
Enter row 2 (space-separated): 6 4
The sum of the two matrices is:
[7, 8]
[9, 8]
```

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)

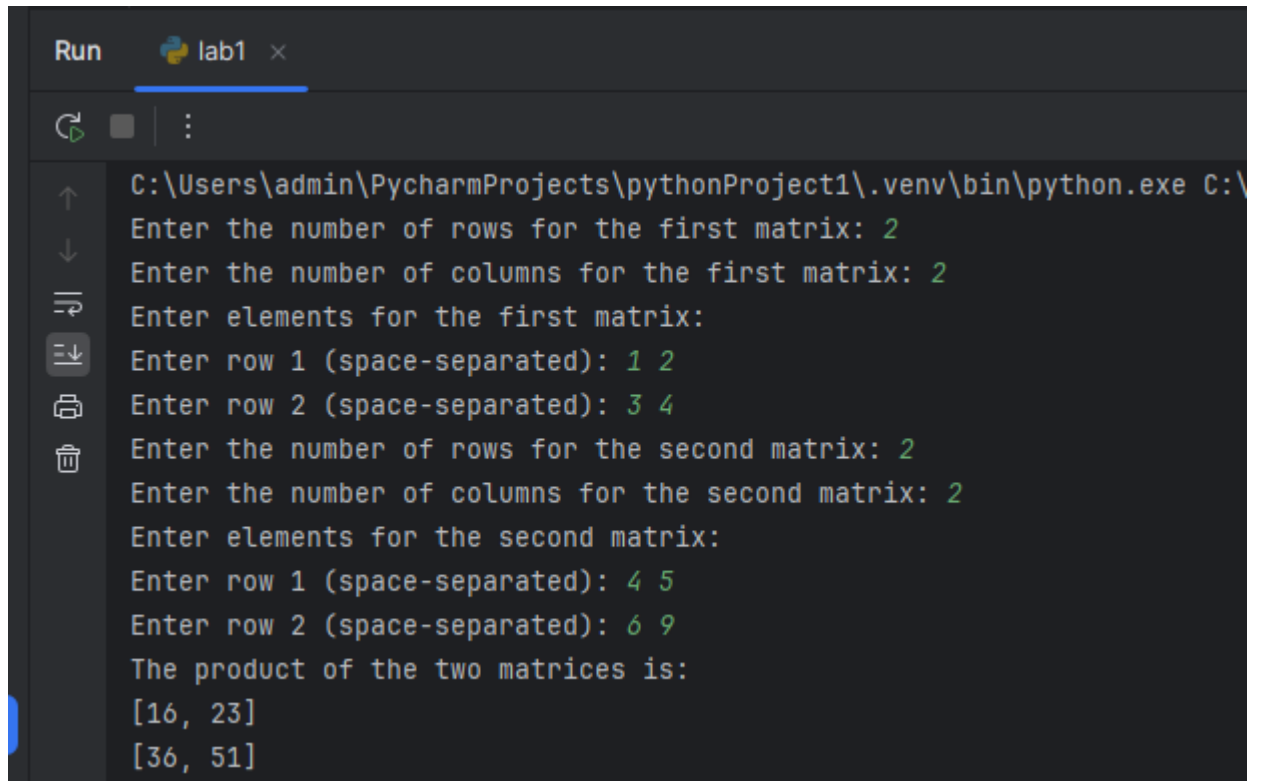
rows2 = int(input("Enter the number of rows for the second matrix: "))
cols2 = int(input("Enter the number of columns for the second matrix: "))

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:



The screenshot shows a PyCharm Run console window for a file named 'lab1'. The console output displays the execution of a Python script that prompts the user to enter the dimensions and elements of two matrices, calculates their product, and prints the result. The first matrix is 2x2 with elements [[1, 2], [3, 4]]. The second matrix is 2x2 with elements [[4, 5], [6, 9]]. The resulting product matrix is [[16, 23], [36, 51]].

```
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\V
Enter the number of rows for the first matrix: 2
Enter the number of columns for the first matrix: 2
Enter elements for the first matrix:
Enter row 1 (space-separated): 1 2
Enter row 2 (space-separated): 3 4
Enter the number of rows for the second matrix: 2
Enter the number of columns for the second matrix: 2
Enter elements for the second matrix:
Enter row 1 (space-separated): 4 5
Enter row 2 (space-separated): 6 9
The product of the two matrices is:
[16, 23]
[36, 51]
```

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:


```
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\ad
Enter an element to append: a
List after appending: ['a']
Enter an element to insert: f
Enter the position to insert (0-based index): 1
List after insertion: ['a', 'f']
Enter an element to remove: f
List after removing 'f': ['a']
Enter elements for another list (space-separated): t y h
The lists are not equal.
First list: ['a']
Second list: ['t', 'y', 'h']
```

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: "))

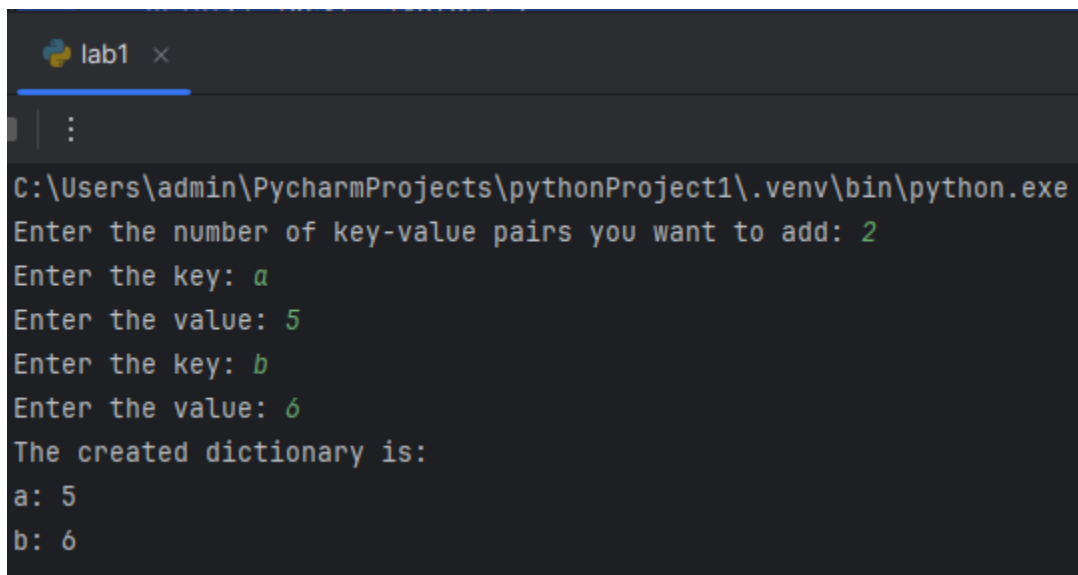
    for _ in range(num_entries):
        key = input("Enter the key: ")
        value = input("Enter the value: ")
        my_dict[key] = value

    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:



```
lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe
Enter the number of key-value pairs you want to add: 2
Enter the key: a
Enter the value: 5
Enter the key: b
Enter the value: 6
The created dictionary is:
a: 5
b: 6
```

Problem 16. Python Program to convert list to dictionary

Python code:

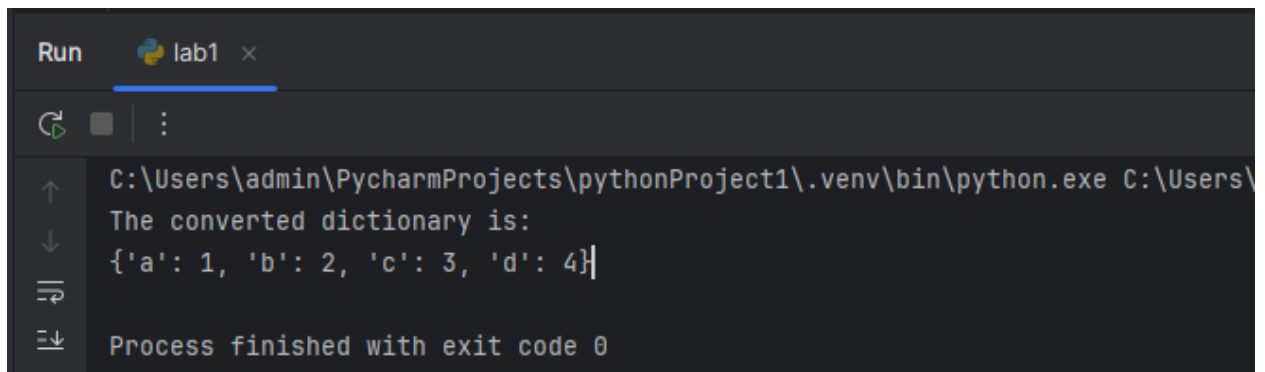
```
# Function to convert a list of tuples to a dictionary
def list_to_dictionary(tuples_list):
    return dict(tuples_list)

# 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:



The screenshot shows a Python IDE window titled "Run" with a sub-tab "lab1". The output console displays the execution of the program. The command prompt shows the path to the Python executable and the script path. The output of the program is "The converted dictionary is:" followed by the dictionary representation of the input list of tuples. The process finished with exit code 0.

```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\
The converted dictionary is:
{'a': 1, 'b': 2, 'c': 3, 'd': 4}
Process finished with exit code 0
```

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:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProjects\pythonProject1\main.py
Original Dictionary:
{'banana': 3, 'apple': 2, 'orange': 5, 'kiwi': 1}

Sorted by Keys:
{'apple': 2, 'banana': 3, 'kiwi': 1, 'orange': 5}

Sorted by Values:
{'kiwi': 1, 'apple': 2, 'banana': 3, 'orange': 5}

Process finished with exit code 0
```

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

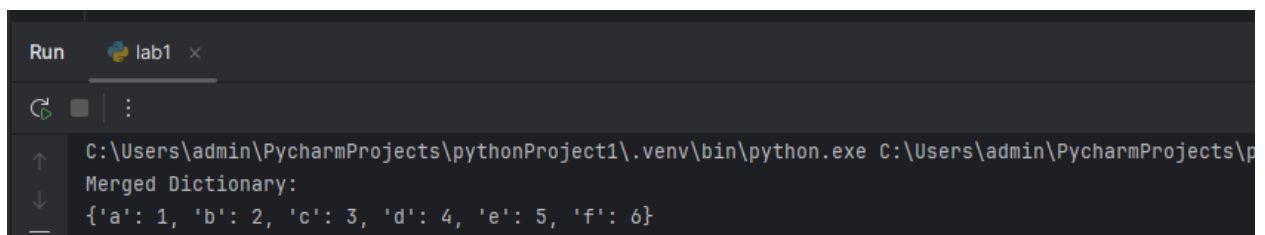
# Main execution
# Example dictionaries
dict1 = {
    'a': 1,
    'b': 2,
    'c': 3
}

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:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\PycharmProjects\p
Merged Dictionary:
{'a': 1, 'b': 2, 'c': 3, 'd': 4, 'e': 5, 'f': 6}
```

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

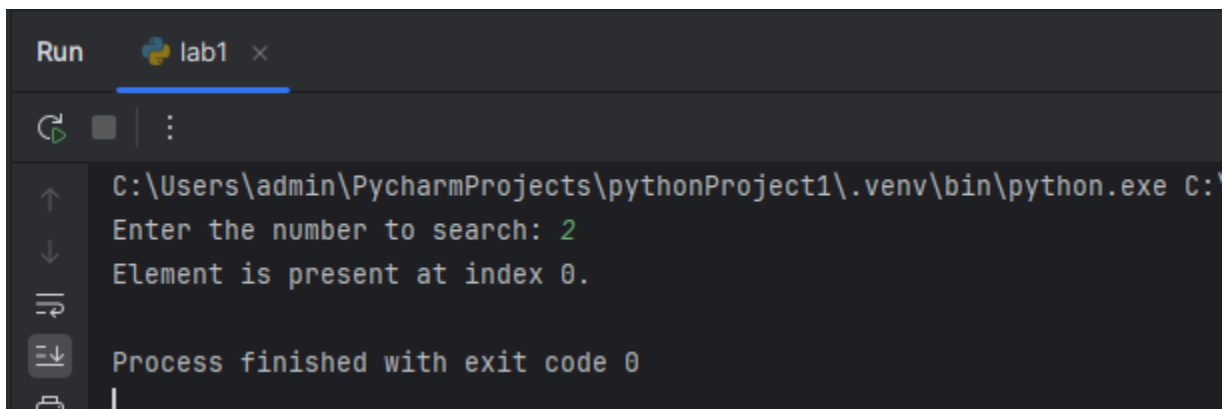
    return -1 # Target not found

# 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:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\
Enter the number to search: 2
Element is present at index 0.
Process finished with exit code 0
```

Problem 20. Linear Search in Python

Python code:

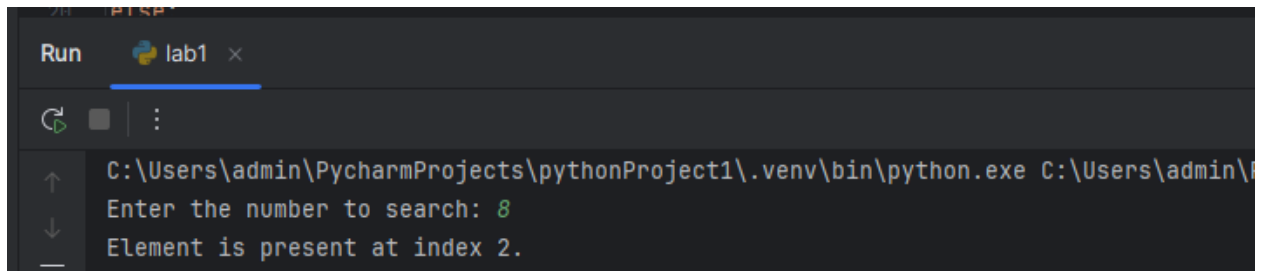
```
# Function to perform linear search
def linear_search(arr, target):
    for index, element in enumerate(arr):
        # Check if the target is present at the current index
        if element == target:
            return index
    return -1 # Target not found

# 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:



```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:\Users\admin\
Enter the number to search: 8
Element is present at index 2.
```

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:

```
Run lab1 x
Original array:
[64, 34, 25, 12, 22, 11, 90]
Sorted array:
[11, 12, 22, 25, 34, 64, 90]
```


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]

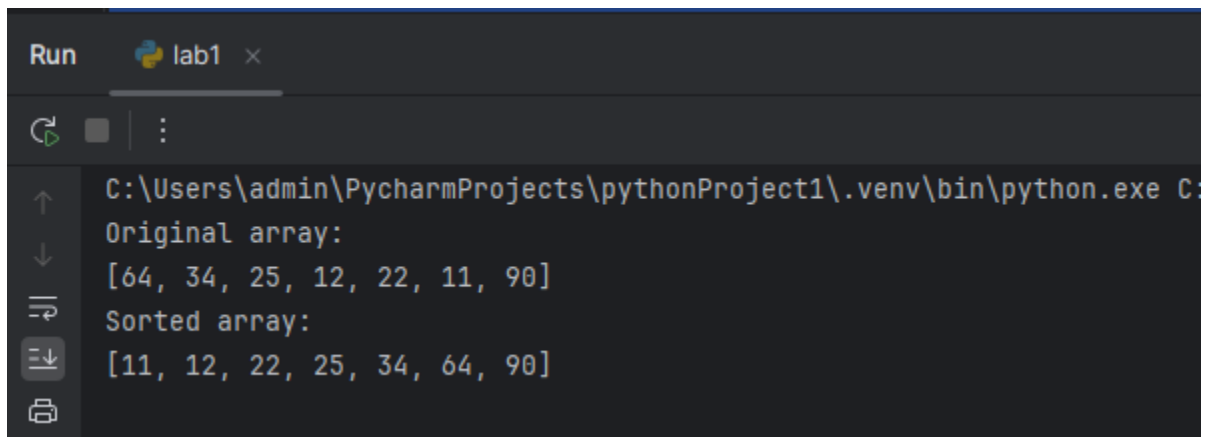
# Main execution
# Example array
arr = [64, 34, 25, 12, 22, 11, 90]

print("Original array:")
print(arr)

# Perform bubble sort
bubble_sort(arr)

# Display the sorted array
print("Sorted array:")
print(arr)
```

Result:

A screenshot of a Python IDE window titled 'Run lab1 x'. The output console shows the execution of the bubble sort function. It starts with the command prompt 'C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:'. The output displays the 'Original array:' as [64, 34, 25, 12, 22, 11, 90] and the 'Sorted array:' as [11, 12, 22, 25, 34, 64, 90]. The IDE interface includes a toolbar with icons for running, stopping, and other actions, as well as a list of files on the left.

```
Run lab1 x
C:\Users\admin\PycharmProjects\pythonProject1\.venv\bin\python.exe C:
Original array:
[64, 34, 25, 12, 22, 11, 90]
Sorted array:
[11, 12, 22, 25, 34, 64, 90]
```