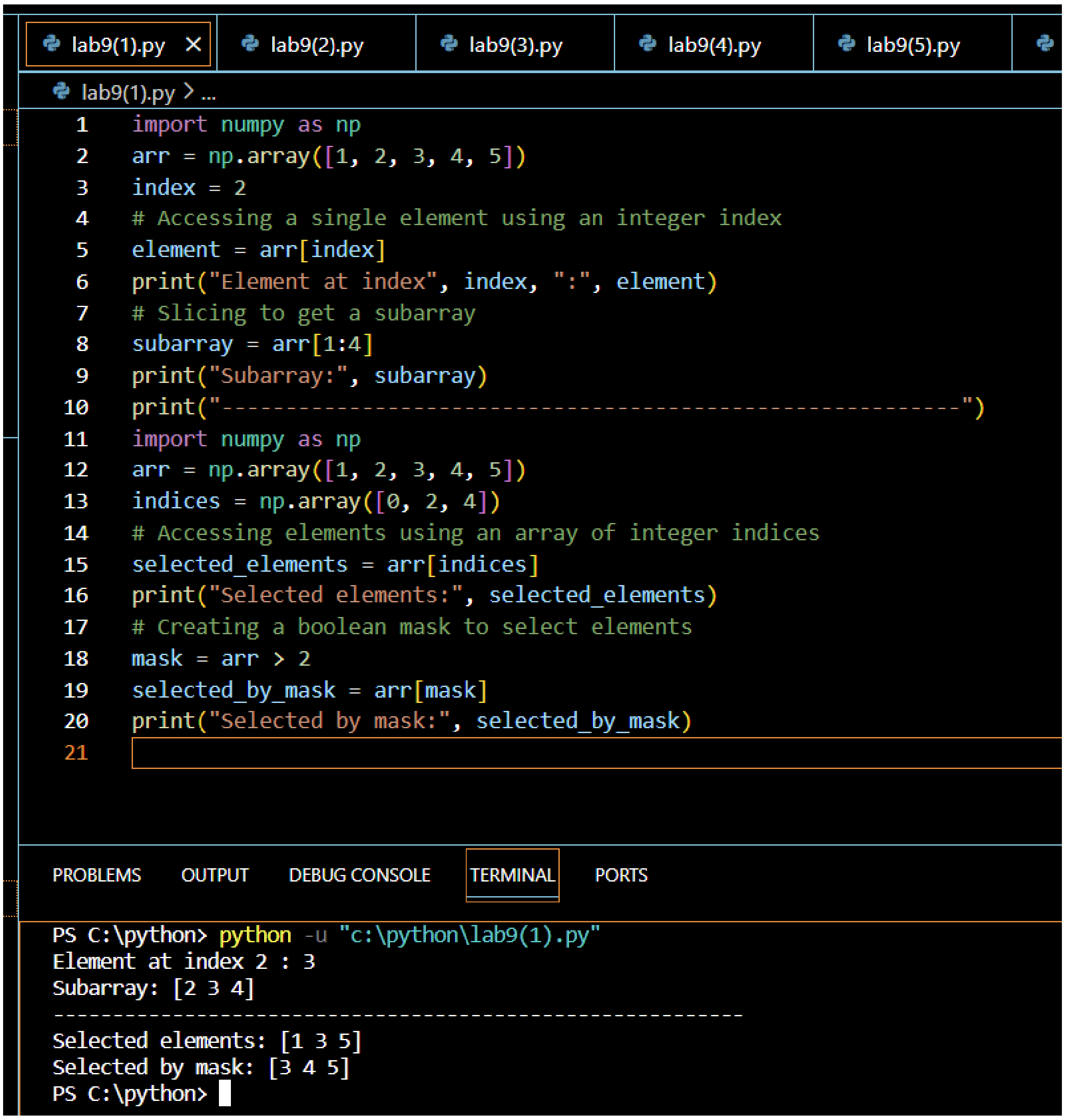
NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

LAB EXERCISE-9

Q1

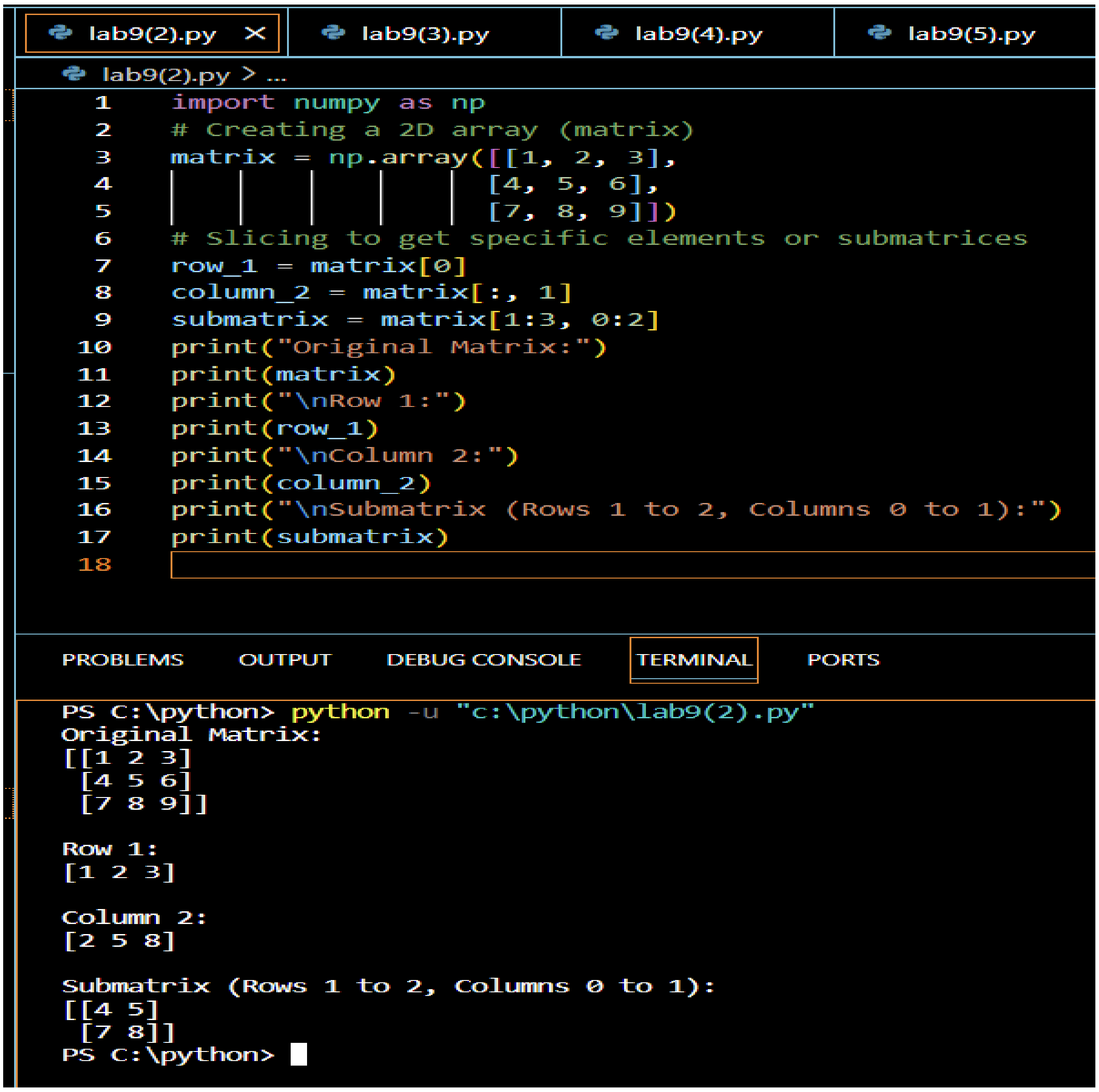


```python
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
index = 2
# Accessing a single element using an integer index
element = arr[index]
print("Element at index", index, ":", element)
# Slicing to get a subarray
subarray = arr[1:4]
print("Subarray:", subarray)
print("----------------------------------------------------------")
import numpy as np
arr = np.array([1, 2, 3, 4, 5])
indices = np.array([0, 2, 4])
# Accessing elements using an array of integer indices
selected_elements = arr[indices]
print("Selected elements:", selected_elements)
# Creating a boolean mask to select elements
mask = arr > 2
selected_by_mask = arr[mask]
print("Selected by mask:", selected_by_mask)
```

```
PS C:\python> python -u "c:\python\lab9(1).py"
Element at index 2 : 3
Subarray: [2 3 4]
----------------------------------------------------------
Selected elements: [1 3 5]
Selected by mask: [3 4 5]
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q2

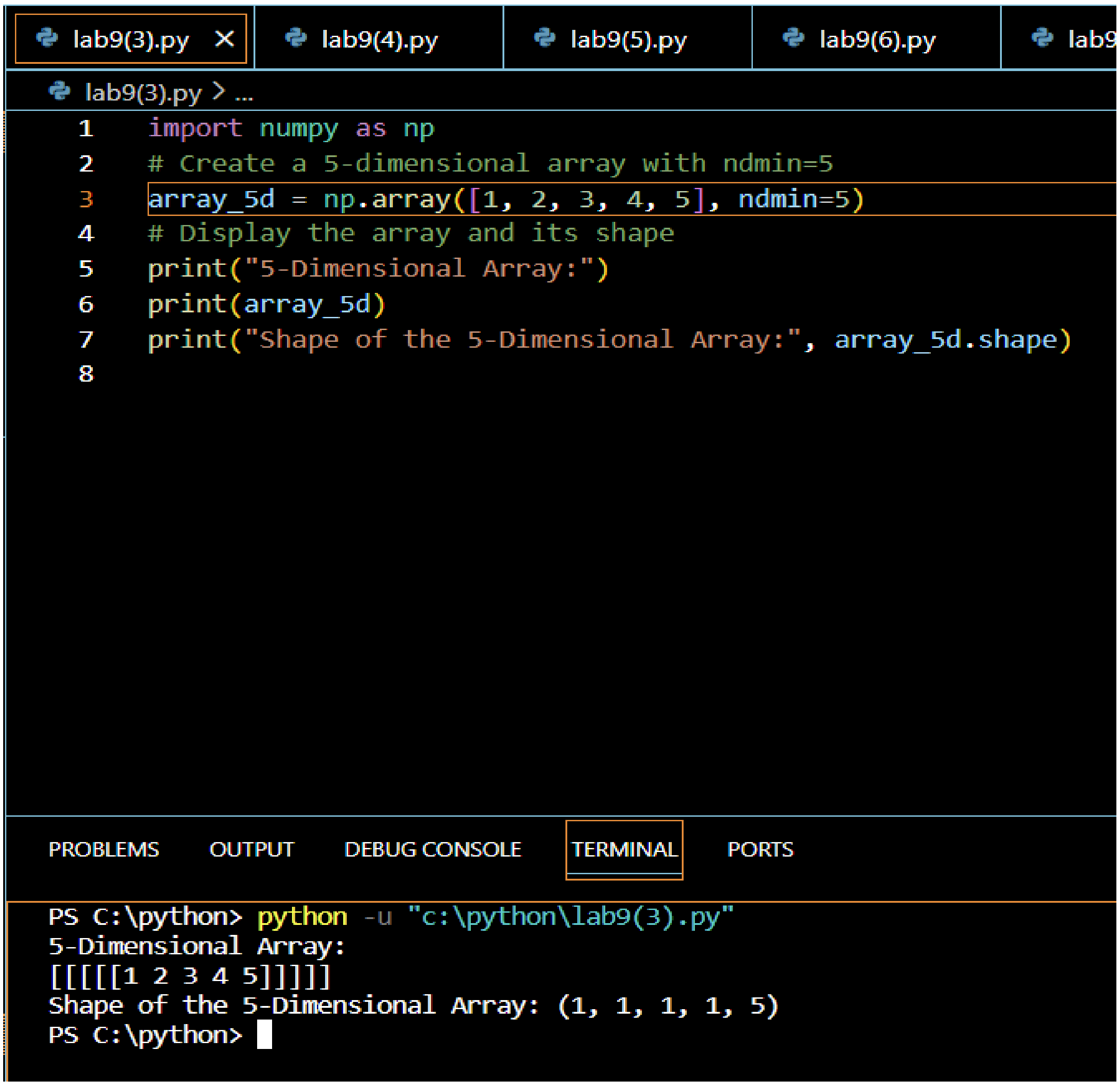


```python
import numpy as np
# Creating a 2D array (matrix)
matrix = np.array([[1, 2, 3],
                   [4, 5, 6],
                   [7, 8, 9]])
# Slicing to get specific elements or submatrices
row_1 = matrix[0]
column_2 = matrix[:, 1]
submatrix = matrix[1:3, 0:2]
print("Original Matrix:")
print(matrix)
print("\nRow 1:")
print(row_1)
print("\nColumn 2:")
print(column_2)
print("\nSubmatrix (Rows 1 to 2, Columns 0 to 1):")
print(submatrix)
```

```
PS C:\python> python -u "c:\python\lab9(2).py"
Original Matrix:
[[1 2 3]
 [4 5 6]
 [7 8 9]]

Row 1:
[1 2 3]

Column 2:
[2 5 8]

Submatrix (Rows 1 to 2, Columns 0 to 1):
[[4 5]
 [7 8]]
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q3

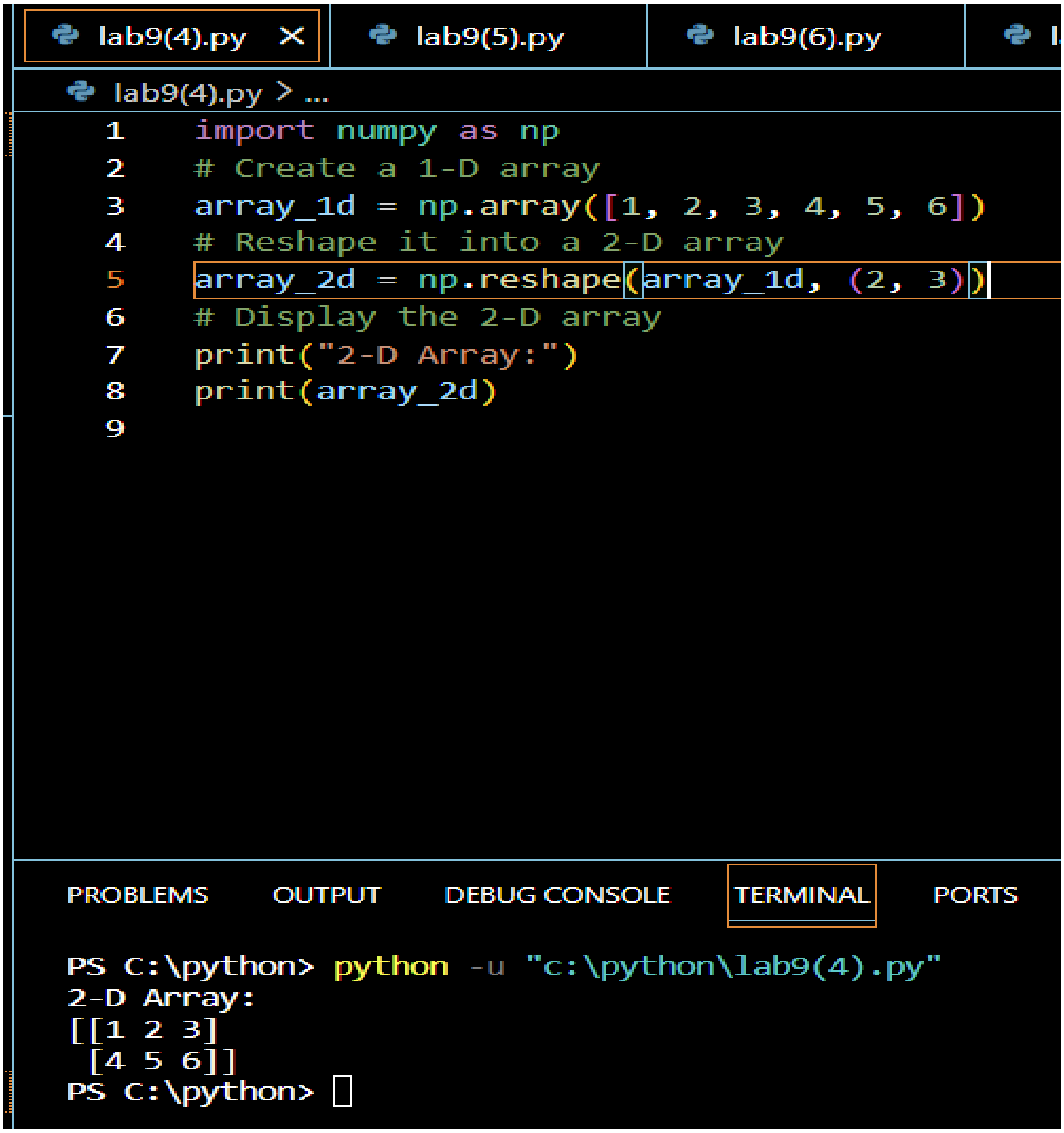


```python
import numpy as np
# Create a 5-dimensional array with ndmin=5
array_5d = np.array([1, 2, 3, 4, 5], ndmin=5)
# Display the array and its shape
print("5-Dimensional Array:")
print(array_5d)
print("Shape of the 5-Dimensional Array:", array_5d.shape)

```

```
PS C:\python> python -u "c:\python\lab9(3).py"
5-Dimensional Array:
[[[[[1 2 3 4 5]]]]]
Shape of the 5-Dimensional Array: (1, 1, 1, 1, 5)
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q4

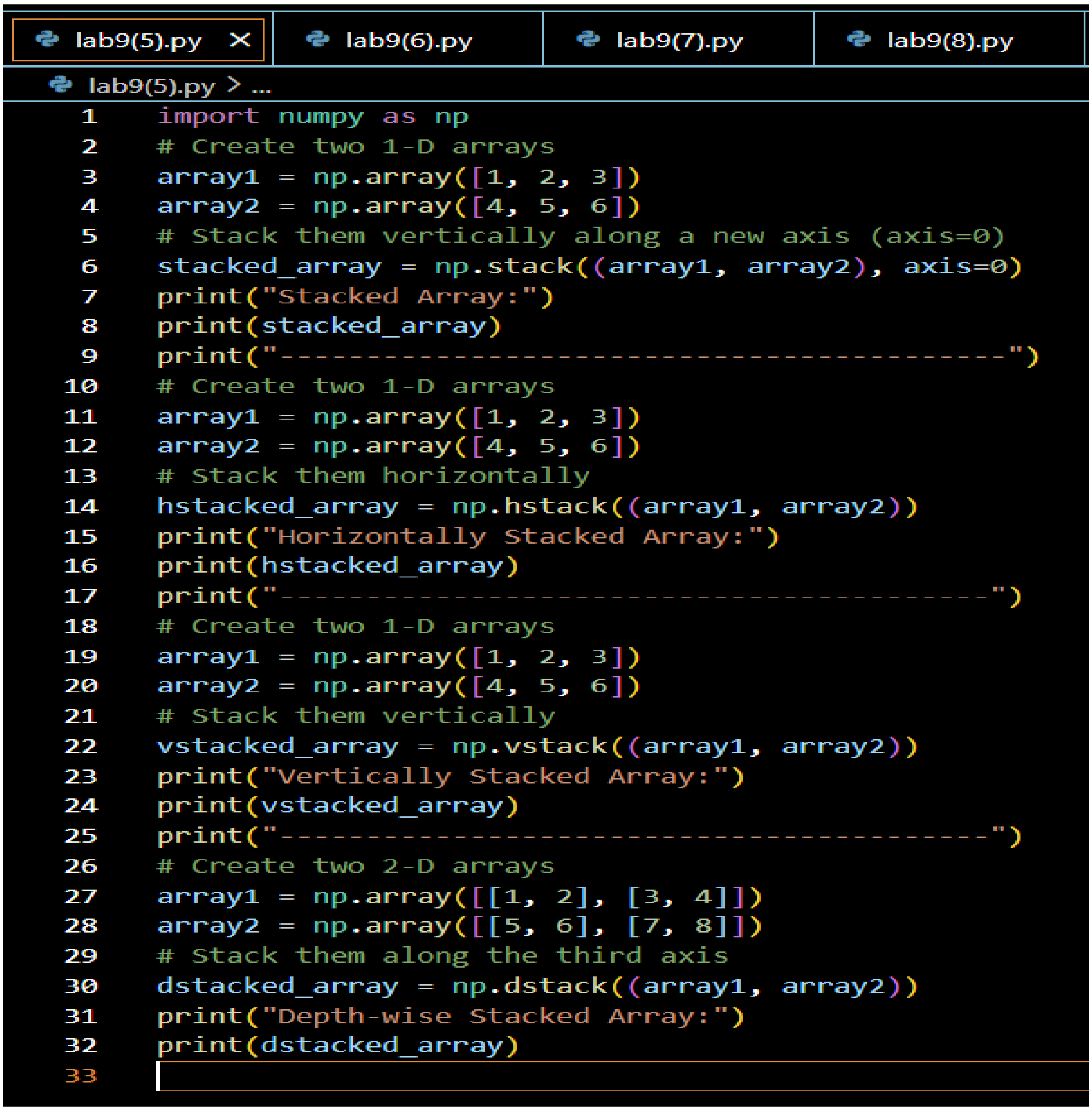


```python
import numpy as np
# Create a 1-D array
array_1d = np.array([1, 2, 3, 4, 5, 6])
# Reshape it into a 2-D array
array_2d = np.reshape(array_1d, (2, 3))
# Display the 2-D array
print("2-D Array:")
print(array_2d)
```

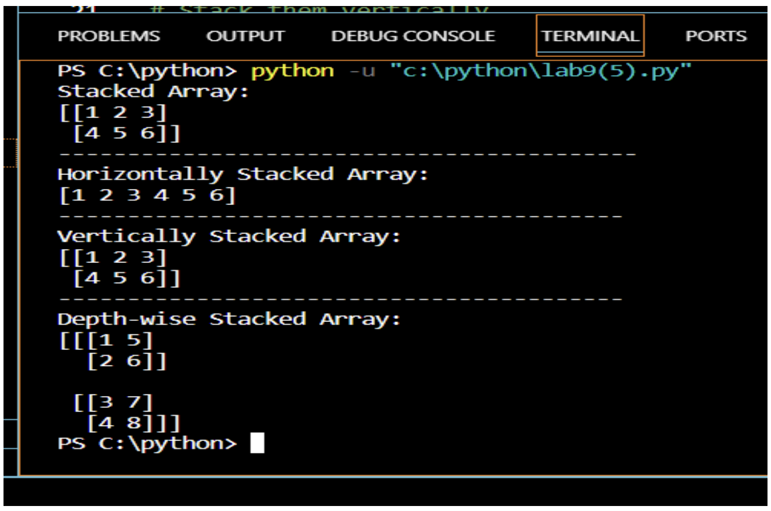
```
PS C:\python> python -u "c:\python\lab9(4).py"
2-D Array:
[[1 2 3]
 [4 5 6]]
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q5



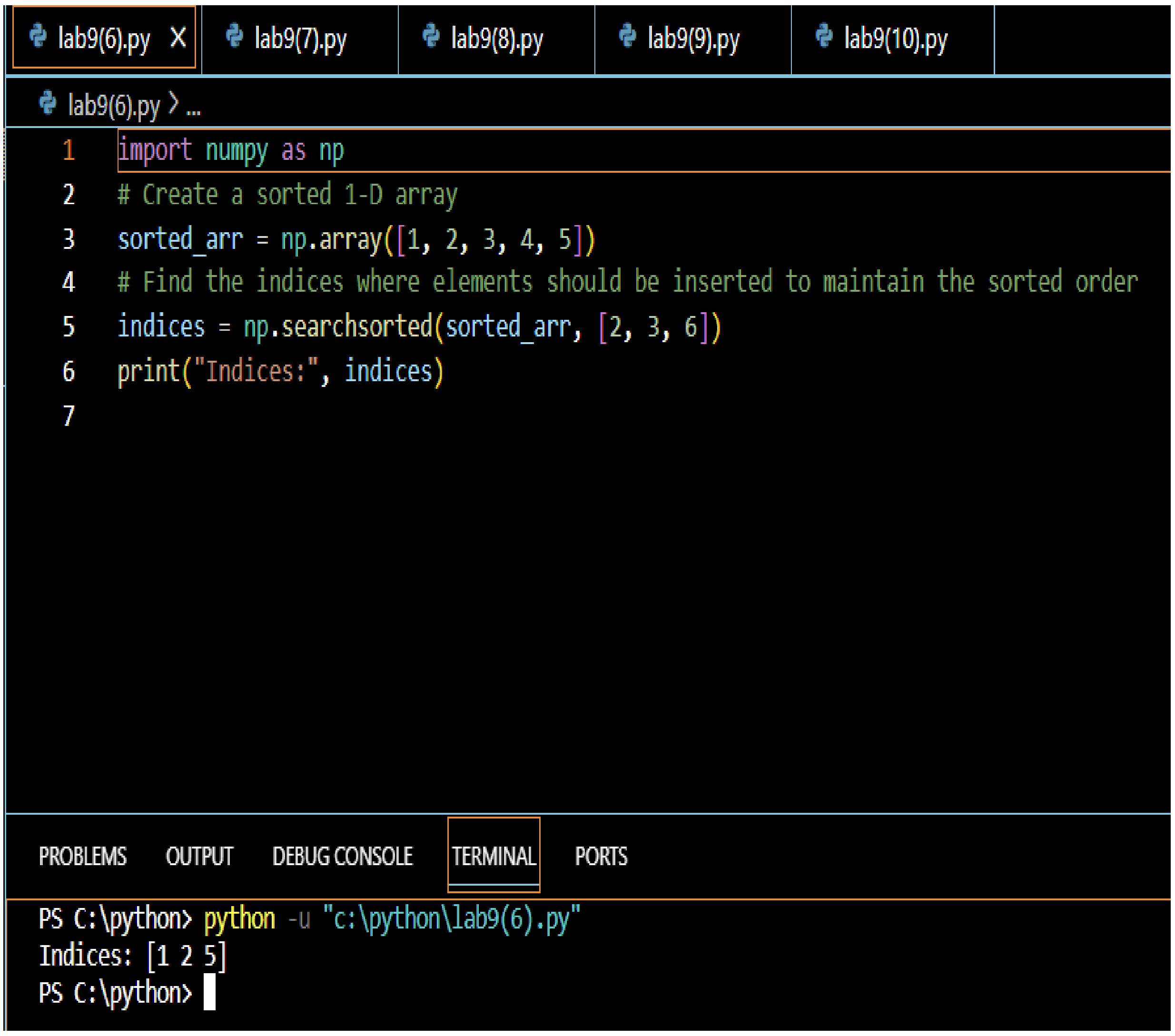
```python
import numpy as np
# Create two 1-D arrays
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
# Stack them vertically along a new axis (axis=0)
stacked_array = np.stack((array1, array2), axis=0)
print("Stacked Array:")
print(stacked_array)
print("-------------------------------------------")
# Create two 1-D arrays
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
# Stack them horizontally
hstacked_array = np.hstack((array1, array2))
print("Horizontally Stacked Array:")
print(hstacked_array)
print("-------------------------------------------")
# Create two 1-D arrays
array1 = np.array([1, 2, 3])
array2 = np.array([4, 5, 6])
# Stack them vertically
vstacked_array = np.vstack((array1, array2))
print("Vertically Stacked Array:")
print(vstacked_array)
print("-------------------------------------------")
# Create two 2-D arrays
array1 = np.array([[1, 2], [3, 4]])
array2 = np.array([[5, 6], [7, 8]])
# Stack them along the third axis
dstacked_array = np.dstack((array1, array2))
print("Depth-wise Stacked Array:")
print(dstacked_array)
```



```
PS C:\python> python -u "c:\python\lab9(5).py"
Stacked Array:
[[1 2 3]
 [4 5 6]]
-------------------------------------------
Horizontally Stacked Array:
[1 2 3 4 5 6]
-------------------------------------------
Vertically Stacked Array:
[[1 2 3]
 [4 5 6]]
-------------------------------------------
Depth-wise Stacked Array:
[[[1 5]
  [2 6]]

 [[3 7]
  [4 8]]]
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q6

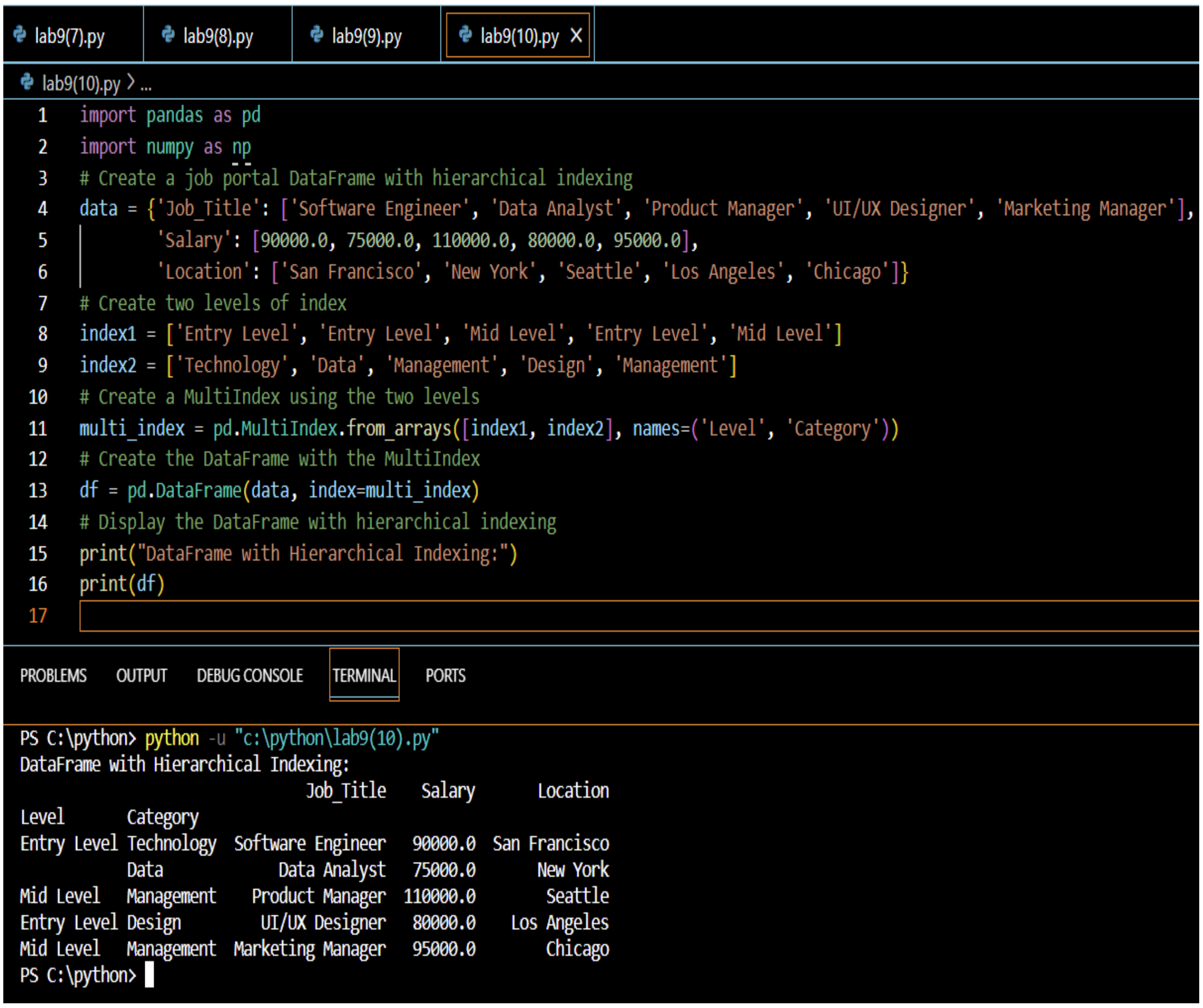


```python
import numpy as np
# Create a sorted 1-D array
sorted_arr = np.array([1, 2, 3, 4, 5])
# Find the indices where elements should be inserted to maintain the sorted order
indices = np.searchsorted(sorted_arr, [2, 3, 6])
print("Indices:", indices)
```

```
PS C:\python> python -u "c:\python\lab9(6).py"
Indices: [1 2 5]
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q7

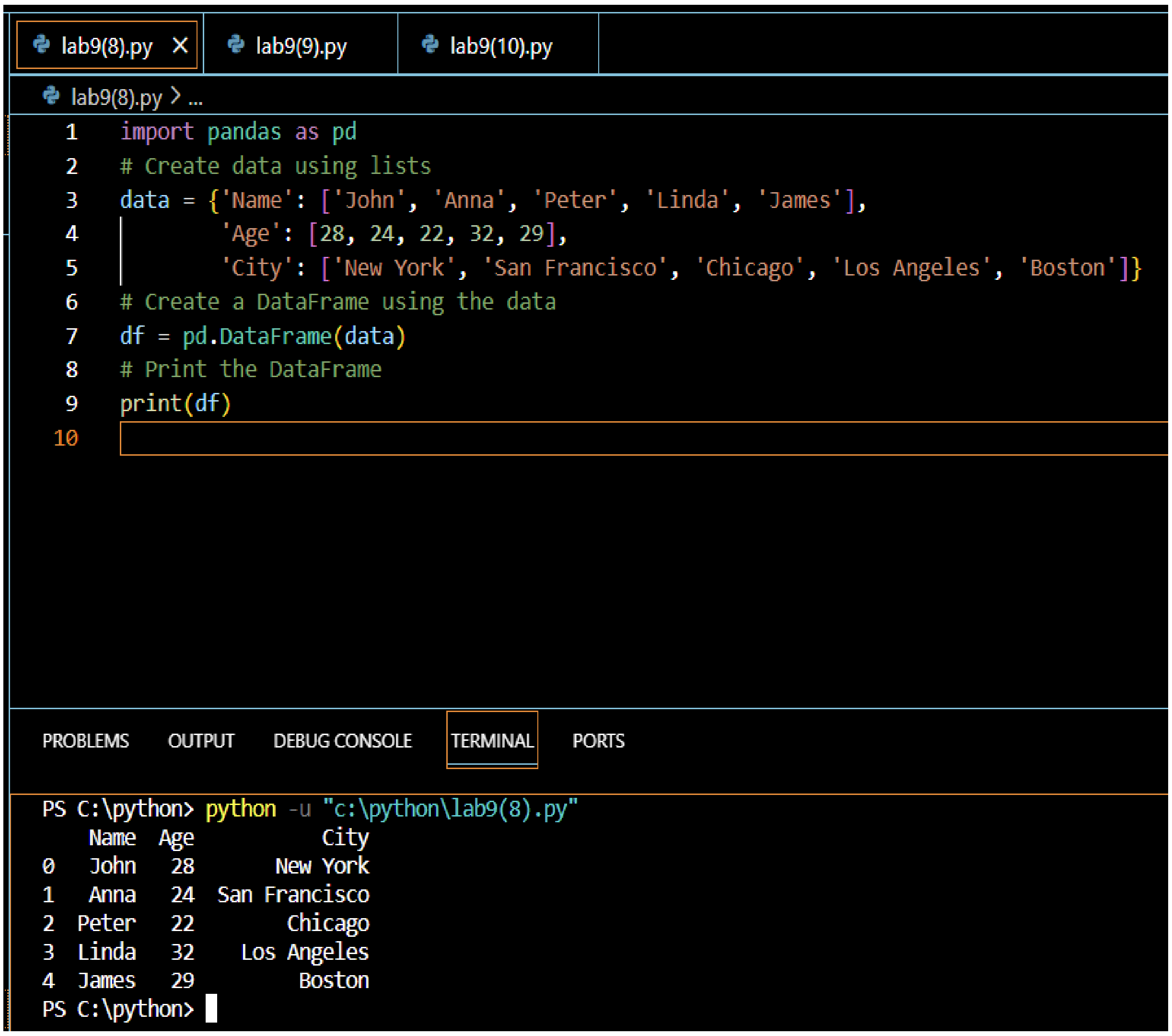


```python
import pandas as pd
import numpy as np
# Create a job portal DataFrame with hierarchical indexing
data = {'Job_Title': ['Software Engineer', 'Data Analyst', 'Product Manager', 'UI/UX Designer', 'Marketing Manager'],
        'Salary': [90000.0, 75000.0, 110000.0, 80000.0, 95000.0],
        'Location': ['San Francisco', 'New York', 'Seattle', 'Los Angeles', 'Chicago']}
# Create two levels of index
index1 = ['Entry Level', 'Entry Level', 'Mid Level', 'Entry Level', 'Mid Level']
index2 = ['Technology', 'Data', 'Management', 'Design', 'Management']
# Create a MultiIndex using the two levels
multi_index = pd.MultiIndex.from_arrays([index1, index2], names=('Level', 'Category'))
# Create the DataFrame with the MultiIndex
df = pd.DataFrame(data, index=multi_index)
# Display the DataFrame with hierarchical indexing
print("DataFrame with Hierarchical Indexing:")
print(df)

```

```
PS C:\python> python -u "c:\python\lab9(10).py"
DataFrame with Hierarchical Indexing:
                                Job_Title    Salary       Location
Level       Category
Entry Level Technology  Software Engineer   90000.0  San Francisco
            Data             Data Analyst   75000.0       New York
Mid Level   Management    Product Manager  110000.0        Seattle
Entry Level Design         UI/UX Designer   80000.0    Los Angeles
Mid Level   Management  Marketing Manager   95000.0        Chicago
PS C:\python> 
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

Q8

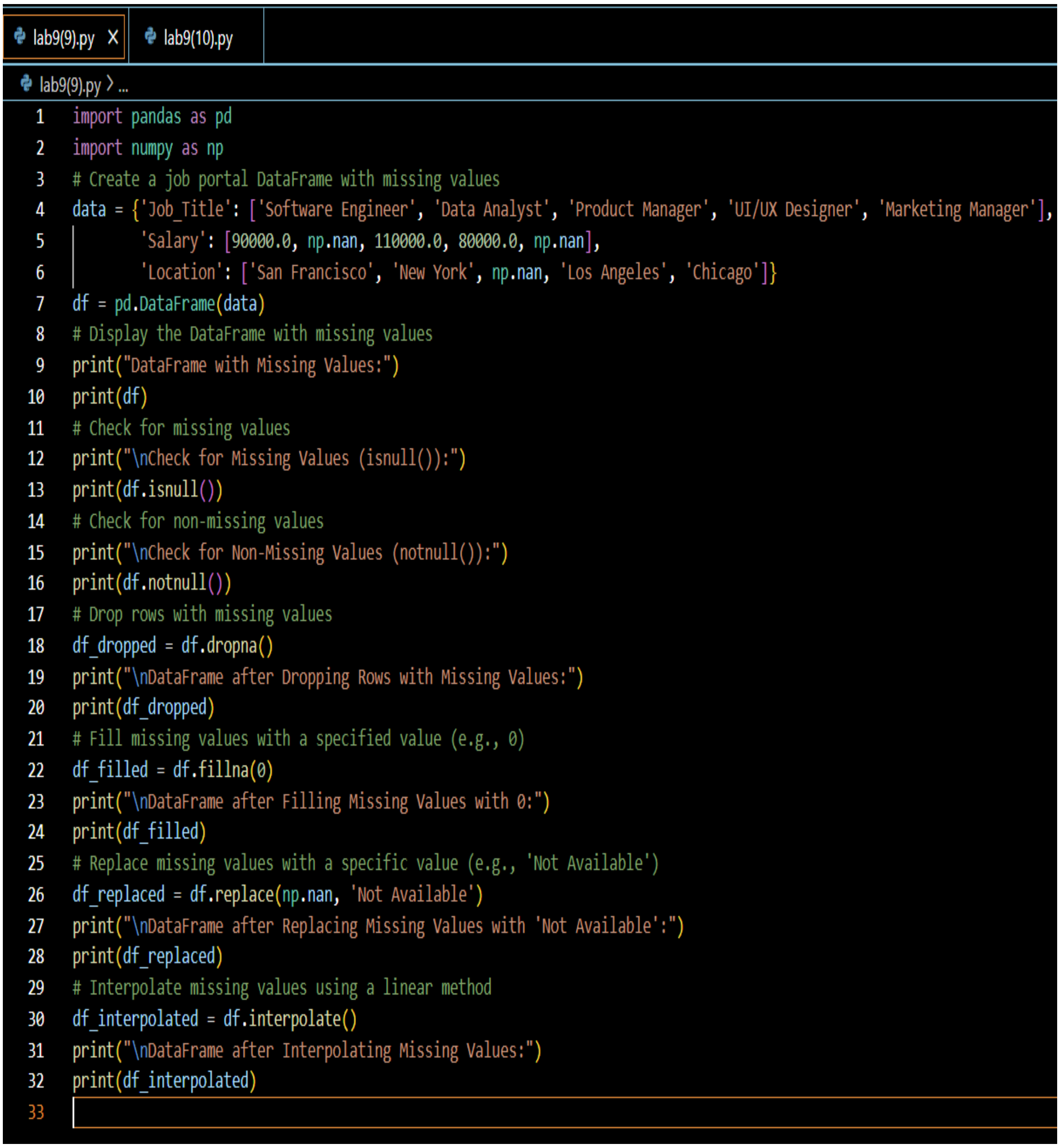


```python
import pandas as pd
# Create data using lists
data = {'Name': ['John', 'Anna', 'Peter', 'Linda', 'James'],
        'Age': [28, 24, 22, 32, 29],
        'City': ['New York', 'San Francisco', 'Chicago', 'Los Angeles', 'Boston']}
# Create a DataFrame using the data
df = pd.DataFrame(data)
# Print the DataFrame
print(df)
```

```
PS C:\python> python -u "c:\python\lab9(8).py"
    Name  Age           City
0   John   28       New York
1   Anna   24  San Francisco
2  Peter   22        Chicago
3  Linda   32    Los Angeles
4  James   29         Boston
PS C:\python>
```

NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255

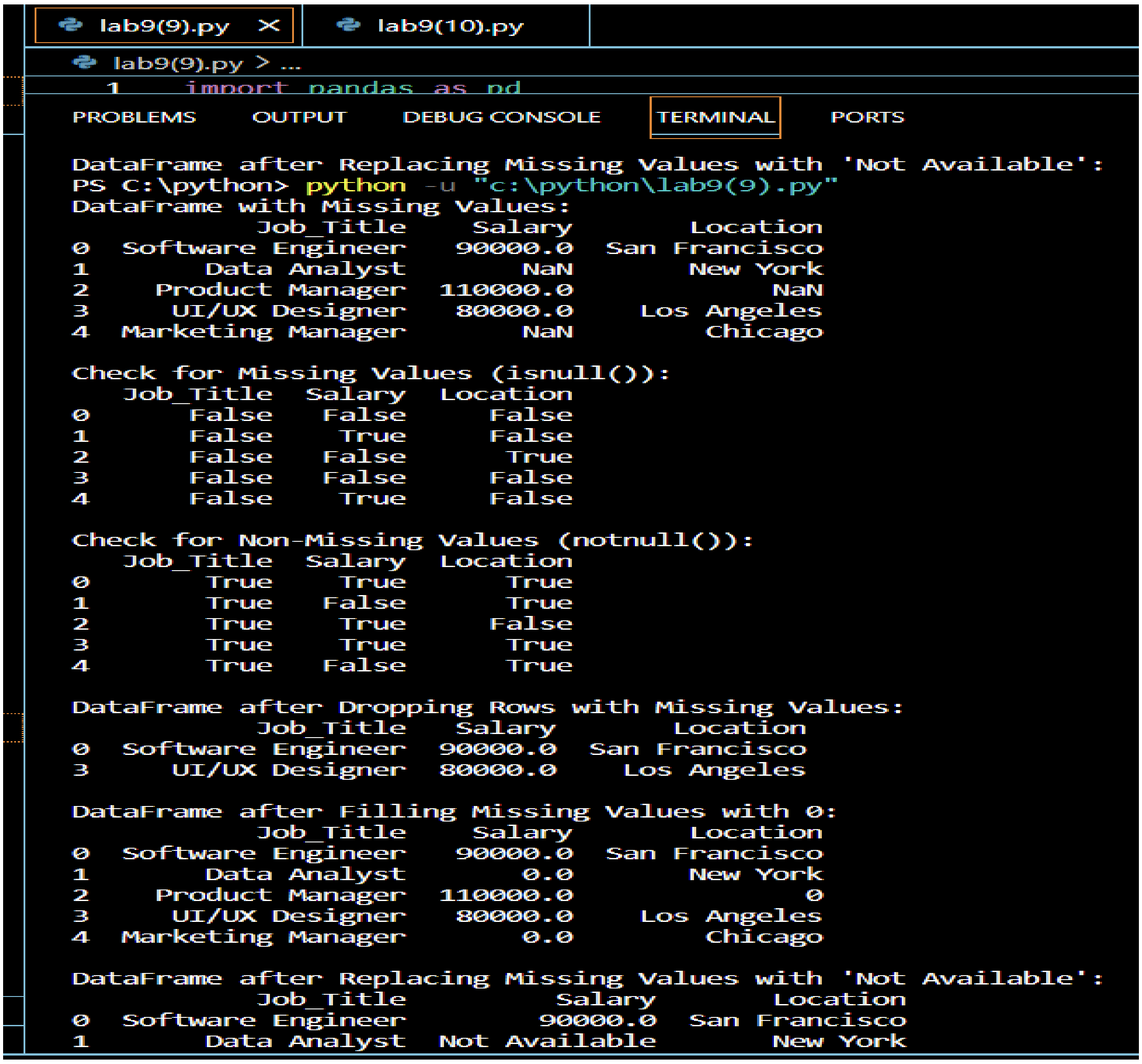
Q9



```python
import pandas as pd
import numpy as np
# Create a job portal DataFrame with missing values
data = {'Job_Title': ['Software Engineer', 'Data Analyst', 'Product Manager', 'UI/UX Designer', 'Marketing Manager'],
        'Salary': [90000.0, np.nan, 110000.0, 80000.0, np.nan],
        'Location': ['San Francisco', 'New York', np.nan, 'Los Angeles', 'Chicago']}
df = pd.DataFrame(data)
# Display the DataFrame with missing values
print("DataFrame with Missing Values:")
print(df)
# Check for missing values
print("\nCheck for Missing Values (isnull()):")
print(df.isnull())
# Check for non-missing values
print("\nCheck for Non-Missing Values (notnull()):")
print(df.notnull())
# Drop rows with missing values
df_dropped = df.dropna()
print("\nDataFrame after Dropping Rows with Missing Values:")
print(df_dropped)
# Fill missing values with a specified value (e.g., 0)
df_filled = df.fillna(0)
print("\nDataFrame after Filling Missing Values with 0:")
print(df_filled)
# Replace missing values with a specific value (e.g., 'Not Available')
df_replaced = df.replace(np.nan, 'Not Available')
print("\nDataFrame after Replacing Missing Values with 'Not Available':")
print(df_replaced)
# Interpolate missing values using a linear method
df_interpolated = df.interpolate()
print("\nDataFrame after Interpolating Missing Values:")
print(df_interpolated)

```

Outputs: -

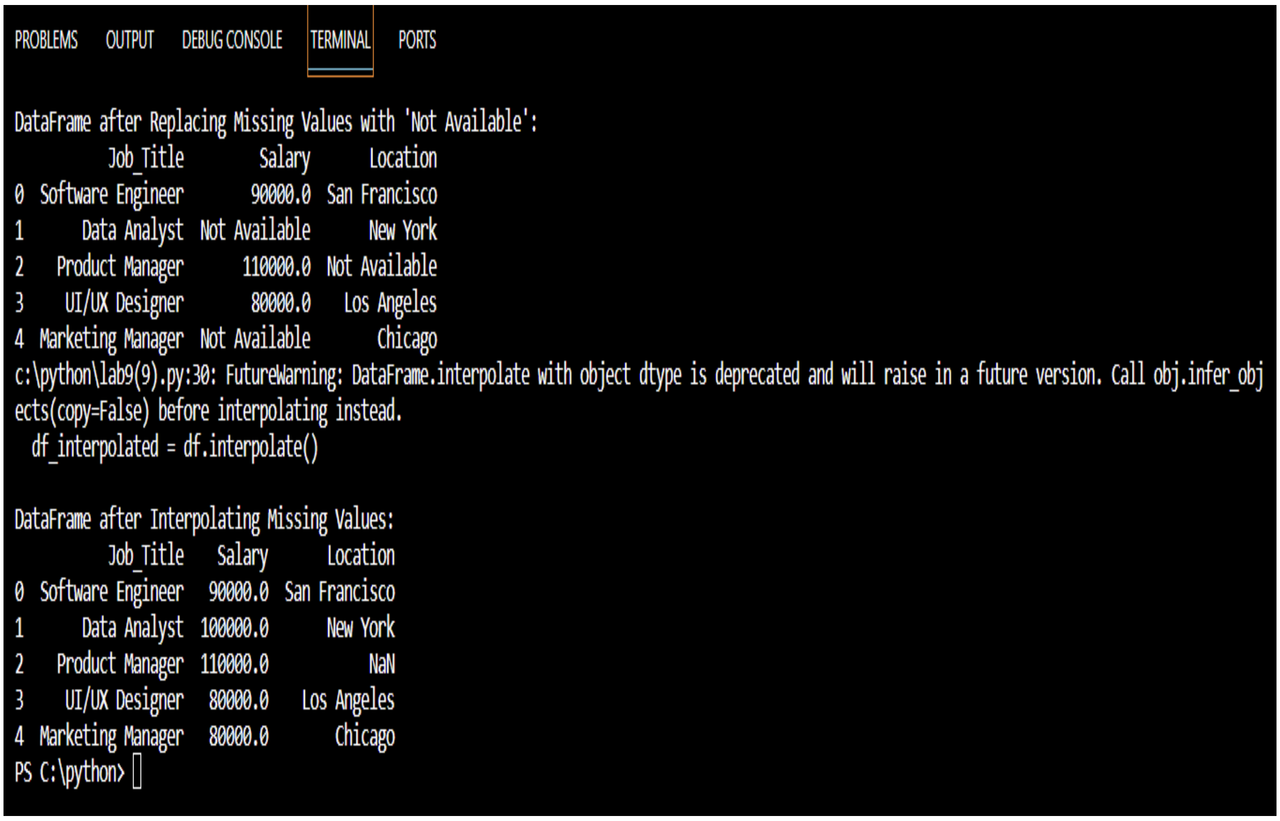
NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255



```
import pandas as pd
```

```
DataFrame after Replacing Missing Values with 'Not Available':
PS C:\python> python -u "c:\python\lab9(9).py"
DataFrame with Missing Values:
           Job_Title    Salary       Location
0  Software Engineer   90000.0  San Francisco
1       Data Analyst       NaN       New York
2    Product Manager  110000.0            NaN
3     UI/UX Designer   80000.0    Los Angeles
4  Marketing Manager       NaN        Chicago

Check for Missing Values (isnull()):
   Job_Title  Salary  Location
0      False   False     False
1      False    True     False
2      False   False      True
3      False   False     False
4      False    True     False

Check for Non-Missing Values (notnull()):
   Job_Title  Salary  Location
0       True    True      True
1       True   False      True
2       True    True     False
3       True    True      True
4       True   False      True

DataFrame after Dropping Rows with Missing Values:
           Job_Title   Salary       Location
0  Software Engineer  90000.0  San Francisco
3     UI/UX Designer  80000.0    Los Angeles

DataFrame after Filling Missing Values with 0:
           Job_Title    Salary       Location
0  Software Engineer   90000.0  San Francisco
1       Data Analyst       0.0       New York
2    Product Manager  110000.0              0
3     UI/UX Designer   80000.0    Los Angeles
4  Marketing Manager       0.0        Chicago

DataFrame after Replacing Missing Values with 'Not Available':
           Job_Title         Salary       Location
0  Software Engineer        90000.0  San Francisco
1       Data Analyst  Not Available       New York
```

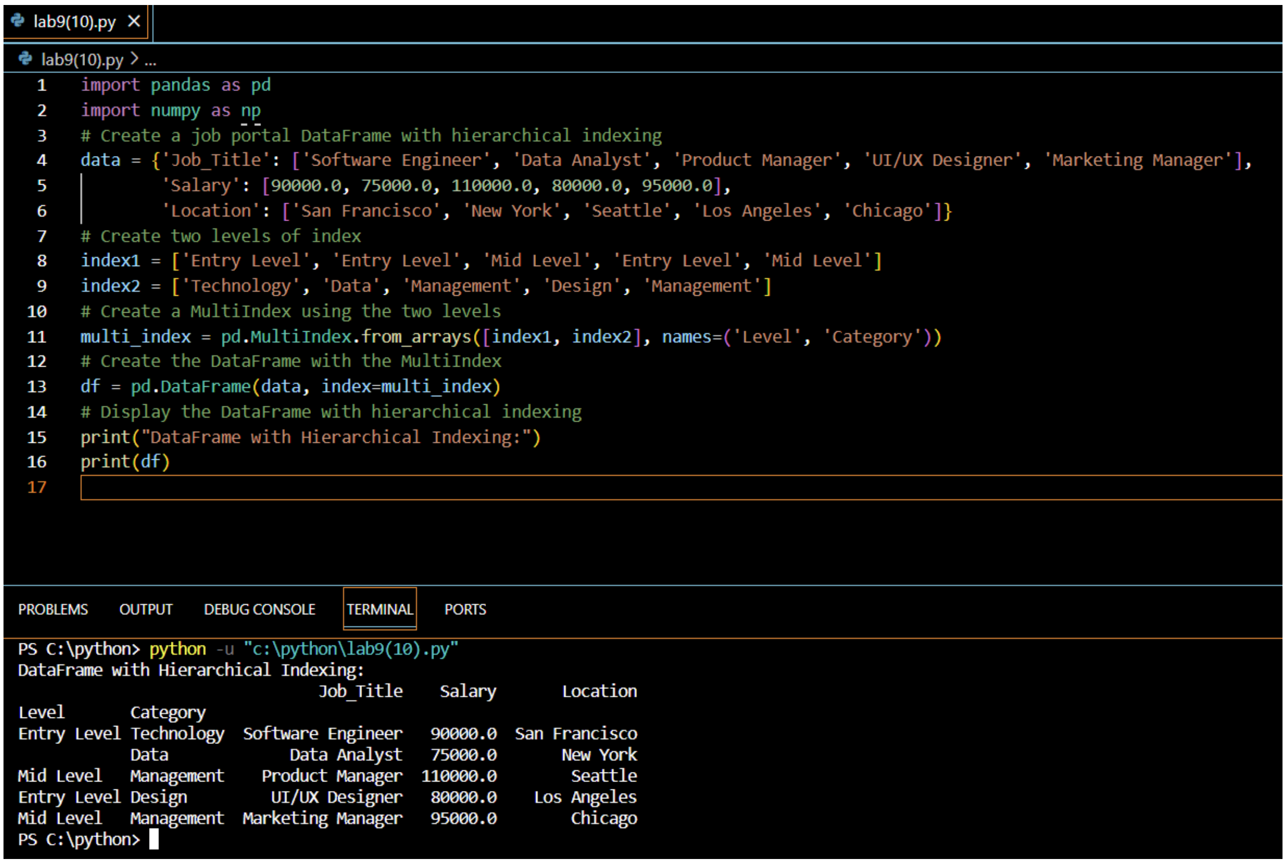
NAME: SHREYA GOEL
CLASS: 1MCA-B
REGNO: 2347255



PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
DataFrame after Replacing Missing Values with 'Not Available':
          Job_Title         Salary       Location
0  Software Engineer        90000.0  San Francisco
1      Data Analyst  Not Available       New York
2   Product Manager       110000.0  Not Available
3     UI/UX Designer        80000.0    Los Angeles
4  Marketing Manager  Not Available        Chicago
c:\python\lab9(9).py:30: FutureWarning: DataFrame.interpolate with object dtype is deprecated and will raise in a future version. Call obj.infer_obj
ects(copy=False) before interpolating instead.
  df_interpolated = df.interpolate()

DataFrame after Interpolating Missing Values:
          Job_Title    Salary       Location
0  Software Engineer   90000.0  San Francisco
1      Data Analyst  100000.0       New York
2   Product Manager  110000.0            NaN
3     UI/UX Designer   80000.0    Los Angeles
4  Marketing Manager   80000.0        Chicago
PS C:\python> 
```

Q10



lab9(10).py ×

lab9(10).py > ...

```python
import pandas as pd
import numpy as np
# Create a job portal DataFrame with hierarchical indexing
data = {'Job_Title': ['Software Engineer', 'Data Analyst', 'Product Manager', 'UI/UX Designer', 'Marketing Manager'],
        'Salary': [90000.0, 75000.0, 110000.0, 80000.0, 95000.0],
        'Location': ['San Francisco', 'New York', 'Seattle', 'Los Angeles', 'Chicago']}
# Create two levels of index
index1 = ['Entry Level', 'Entry Level', 'Mid Level', 'Entry Level', 'Mid Level']
index2 = ['Technology', 'Data', 'Management', 'Design', 'Management']
# Create a MultiIndex using the two levels
multi_index = pd.MultiIndex.from_arrays([index1, index2], names=('Level', 'Category'))
# Create the DataFrame with the MultiIndex
df = pd.DataFrame(data, index=multi_index)
# Display the DataFrame with hierarchical indexing
print("DataFrame with Hierarchical Indexing:")
print(df)

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\python> python -u "c:\python\lab9(10).py"
DataFrame with Hierarchical Indexing:
                                 Job_Title    Salary       Location
Level       Category
Entry Level Technology  Software Engineer   90000.0  San Francisco
            Data             Data Analyst   75000.0       New York
Mid Level   Management    Product Manager  110000.0        Seattle
Entry Level Design         UI/UX Designer   80000.0    Los Angeles
Mid Level   Management  Marketing Manager   95000.0        Chicago
PS C:\python> 
```