

PRACTICAL-6

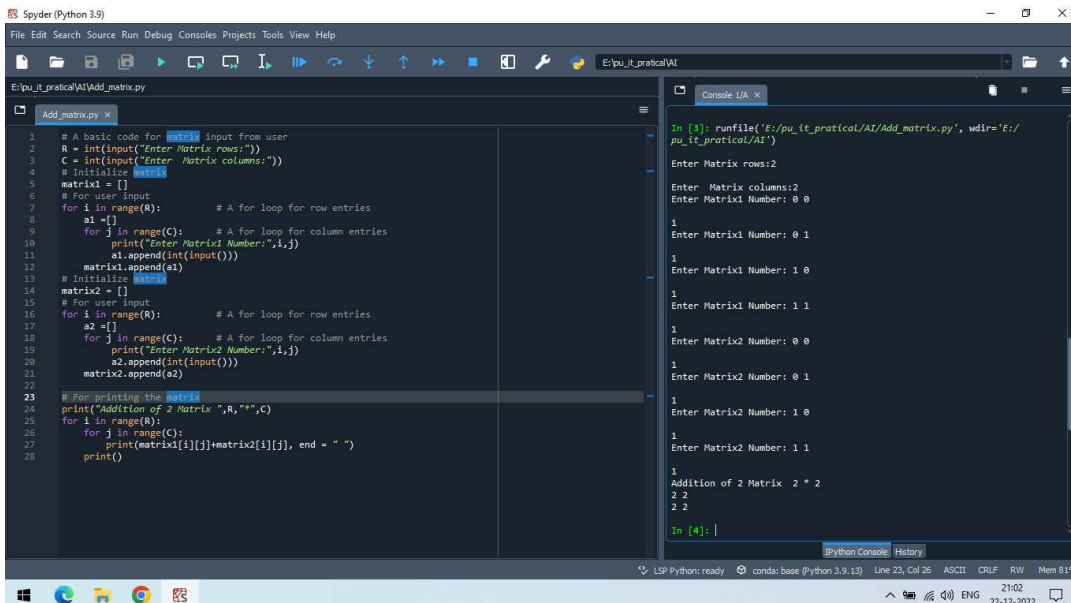
AIM: a) Write a python program to Add Two Matrices.

b) Write a python program to Transpose a Matrix.

Code a)

```
R = int(input("Enter Matrix rows:"))
C = int(input("Enter Matrix columns:"))
matrix1 = [] # Initialize matrix
# For user input
for i in range(R): # A for loop for row entries
    a1 = []
    for j in range(C): # A for loop for column entries
        print("Enter Matrix1 Number:",i,j)
        a1.append(int(input()))
    matrix1.append(a1)
matrix2 = [] # Initialize matrix
# For user input
for i in range(R): # A for loop for row entries
    a2 = []
    for j in range(C): # A for loop for column entries
        print("Enter Matrix2 Number:",i,j)
        a2.append(int(input()))
    matrix2.append(a2)
print("Addition of 2 Matrix ",R,"*",C) # For printing the matrix
for i in range(R):
    for j in range(C):
        print(matrix1[i][j]+matrix2[i][j], end = " ")
    print()
```

Output:



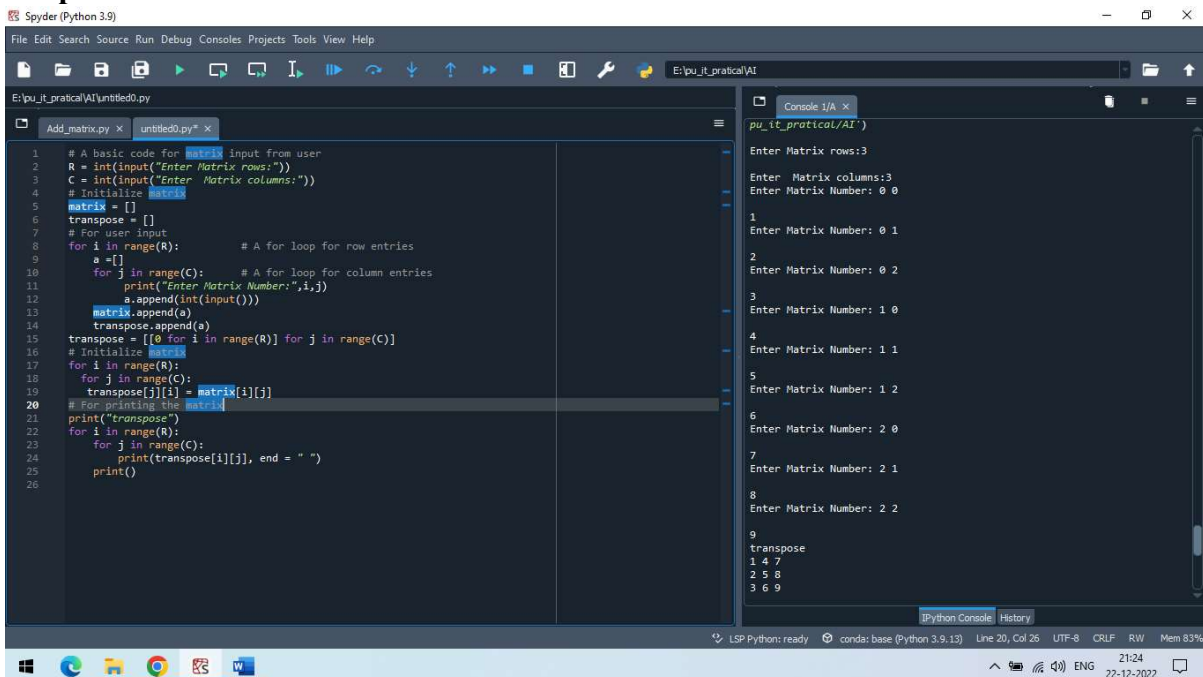
```
# Spyder (Python 3.9)
File Edit Search Source Run Debug Consoles Projects Tools View Help
E:\pu_it_practical\AI\Add_matrix.py
Add_matrix.py
1 # A basic code for matrix input from user
2 R = int(input("Enter Matrix rows:"))
3 C = int(input("Enter Matrix columns:"))
4 # Initialize matrix
5 matrix1 = []
6 # For user input
7 for i in range(R): # A for loop for row entries
8     a1 = []
9     for j in range(C): # A for loop for column entries
10        print("Enter Matrix1 Number:",i,j)
11        a1.append(int(input()))
12    matrix1.append(a1)
13 # Initialize matrix
14 matrix2 = []
15 # For user input
16 for i in range(R): # A for loop for row entries
17     a2 = []
18     for j in range(C): # A for loop for column entries
19        print("Enter Matrix2 Number:",i,j)
20        a2.append(int(input()))
21    matrix2.append(a2)
22
23 # For printing the matrix
24 print("Addition of 2 Matrix ",R,"*",C)
25 for i in range(R):
26     for j in range(C):
27        print(matrix1[i][j]+matrix2[i][j], end = " ")
28     print()

Console 1/A x
In [3]: runfile('E:/pu_it_practical/AI/Add_matrix.py', wdir='E:/pu_it_practical/AI')
Enter Matrix rows:2
Enter Matrix columns:2
Enter Matrix1 Number: 0 0
1
Enter Matrix1 Number: 0 1
1
Enter Matrix1 Number: 1 0
1
Enter Matrix1 Number: 1 1
1
Enter Matrix2 Number: 0 0
1
Enter Matrix2 Number: 0 1
1
Enter Matrix2 Number: 1 0
1
Enter Matrix2 Number: 1 1
1
Addition of 2 Matrix  2 * 2
2 2
2 2
In [4]:
```

Code b)

```
R = int(input("Enter Matrix rows:"))
C = int(input("Enter Matrix columns:"))
matrix = [] # Initialize matrix
transpose = []
# For user input
for i in range(R): # A for loop for row entries
    a = []
    for j in range(C): # A for loop for column entries
        print("Enter Matrix Number:", i, j)
        a.append(int(input()))
    matrix.append(a)
    transpose.append(a)
transpose = [[0 for i in range(R)] for j in range(C)]
# Initialize matrix
for i in range(R):
    for j in range(C):
        transpose[j][i] = matrix[i][j]
# For printing the matrix
print("transpose")
for i in range(R):
    for j in range(C):
        print(transpose[i][j], end = " ")
    print()
```

Output:



The screenshot shows the Spyder Python IDE with the code from the previous block on the left and the output on the right. The output shows the user input for rows (3) and columns (3), followed by a loop where the user enters matrix elements. The final output displays the transpose of the matrix.

```
pu_it_practical/AI
Enter Matrix rows:3
Enter Matrix columns:3
Enter Matrix Number: 0 0
1
Enter Matrix Number: 0 1
2
Enter Matrix Number: 0 2
3
Enter Matrix Number: 1 0
4
Enter Matrix Number: 1 1
5
Enter Matrix Number: 1 2
6
Enter Matrix Number: 2 0
7
Enter Matrix Number: 2 1
8
Enter Matrix Number: 2 2
9
transpose
1 4 7
2 5 8
3 6 9
```