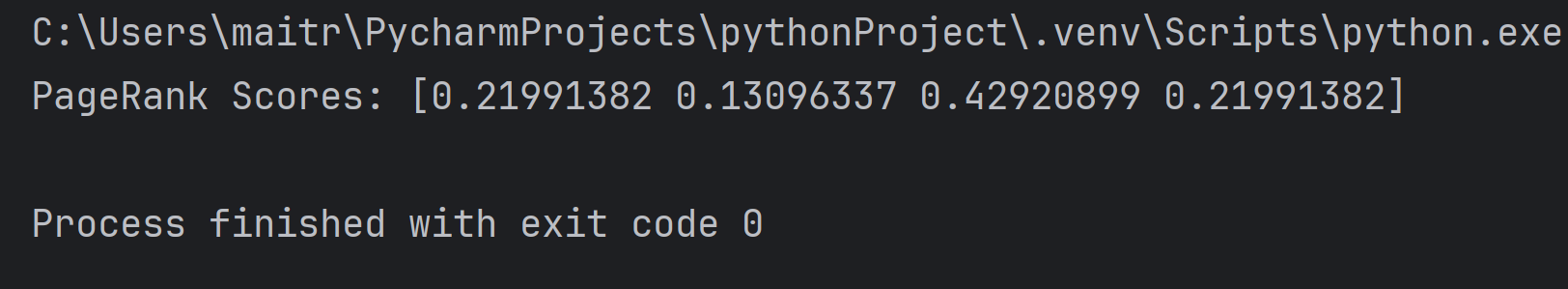
**PRACTICAL 9**

**AIM : page rank**

**INPUT:**

import numpy as np  
def pagerank(M, num\_iterations: int = 100, d: float = 0.85):  
   
 # Number of pages  
 N = M.shape[1]  
  
 # Create a column vector for pagerank values and initialize with 1/N  
 pagerank\_values = np.ones(N) / N  
  
 # Calculate the out-degree for each page  
 out\_degree = np.sum(M, axis=0)  
  
 # Perform the iterations  
 for i in range(num\_iterations):  
 new\_pagerank = np.zeros(N)  
  
 # Update each pagerank value based on incoming links  
 for j in range(N):  
 for i in range(N):  
 if M[i, j] == 1:  
 new\_pagerank[i] += pagerank\_values[j] / out\_degree[j]  
  
 # Apply the damping factor  
 pagerank\_values = d \* new\_pagerank + (1 - d) / N  
  
 return pagerank\_values  
  
# Example usage:  
  
# Adjacency matrix where M[i, j] is 1 if page j links to page i  
M = np.array([[0, 0, 1, 0],  
 [1, 0, 0, 0],  
 [1, 1, 0, 1],  
 [0, 0, 1, 0]])  
  
# Run pagerank  
pagerank\_scores = pagerank(M)  
  
print("PageRank Scores:", pagerank\_scores)

** OUTPUT:**