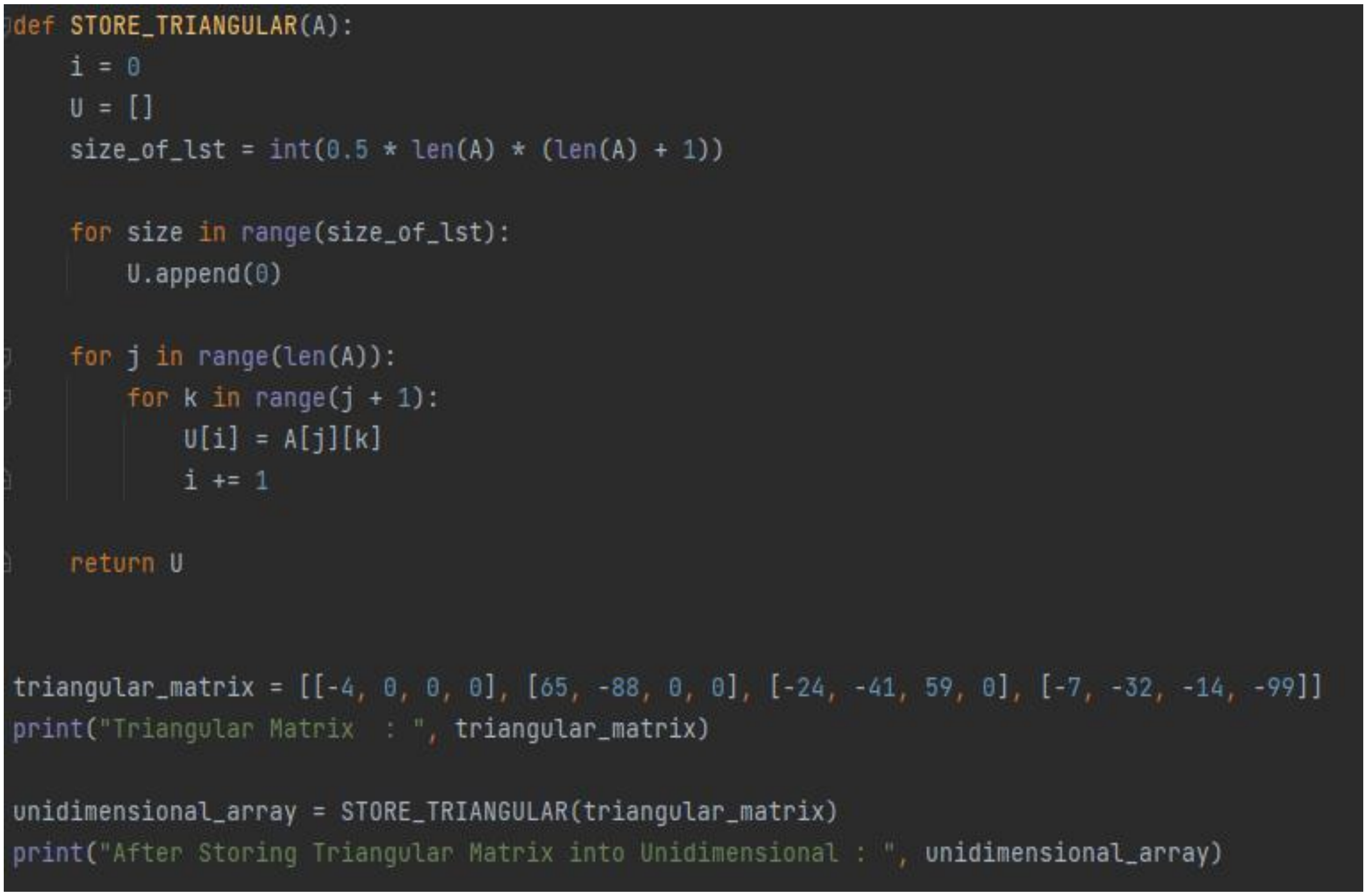
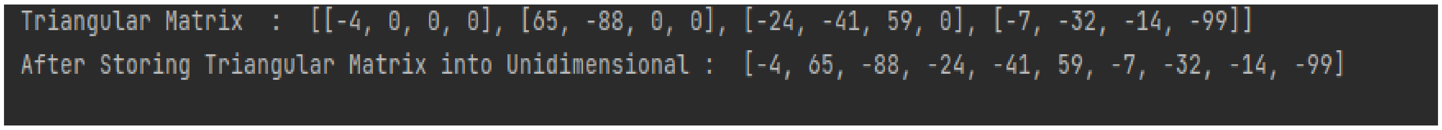
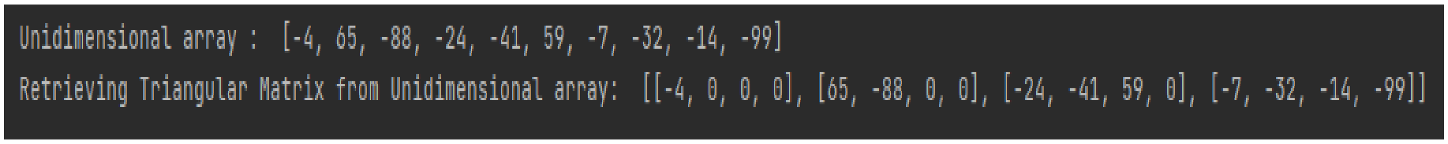


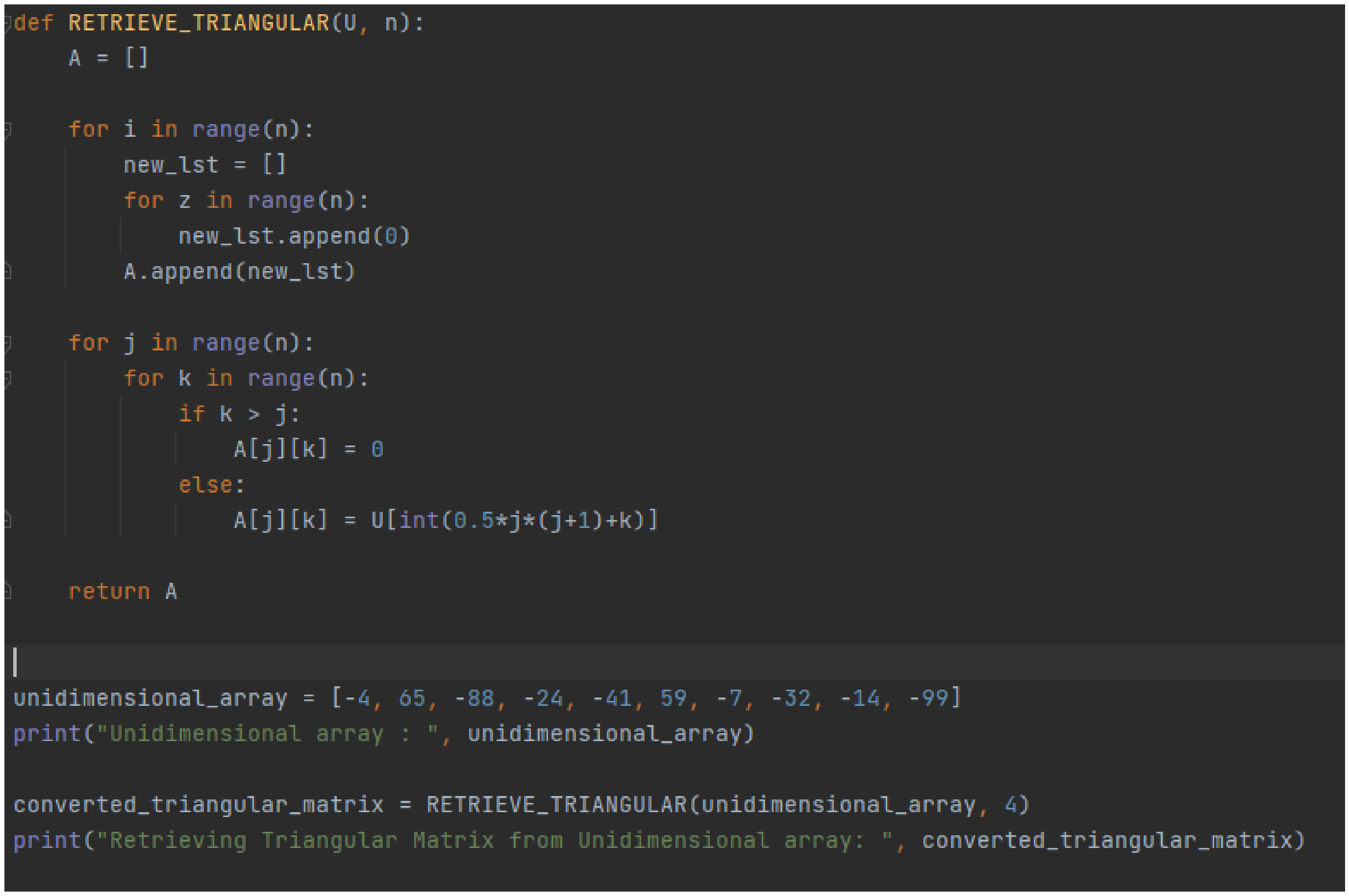
```python
def STORE_TRIANGULAR(A):
    i = 0
    U = []
    size_of_lst = int(0.5 * len(A) * (len(A) + 1))

    for size in range(size_of_lst):
        U.append(0)

    for j in range(len(A)):
        for k in range(j + 1):
            U[i] = A[j][k]
            i += 1

    return U


triangular_matrix = [[-4, 0, 0, 0], [65, -88, 0, 0], [-24, -41, 59, 0], [-7, -32, -14, -99]]
print("Triangular Matrix  : ", triangular_matrix)

unidimensional_array = STORE_TRIANGULAR(triangular_matrix)
print("After Storing Triangular Matrix into Unidimensional : ", unidimensional_array)
```



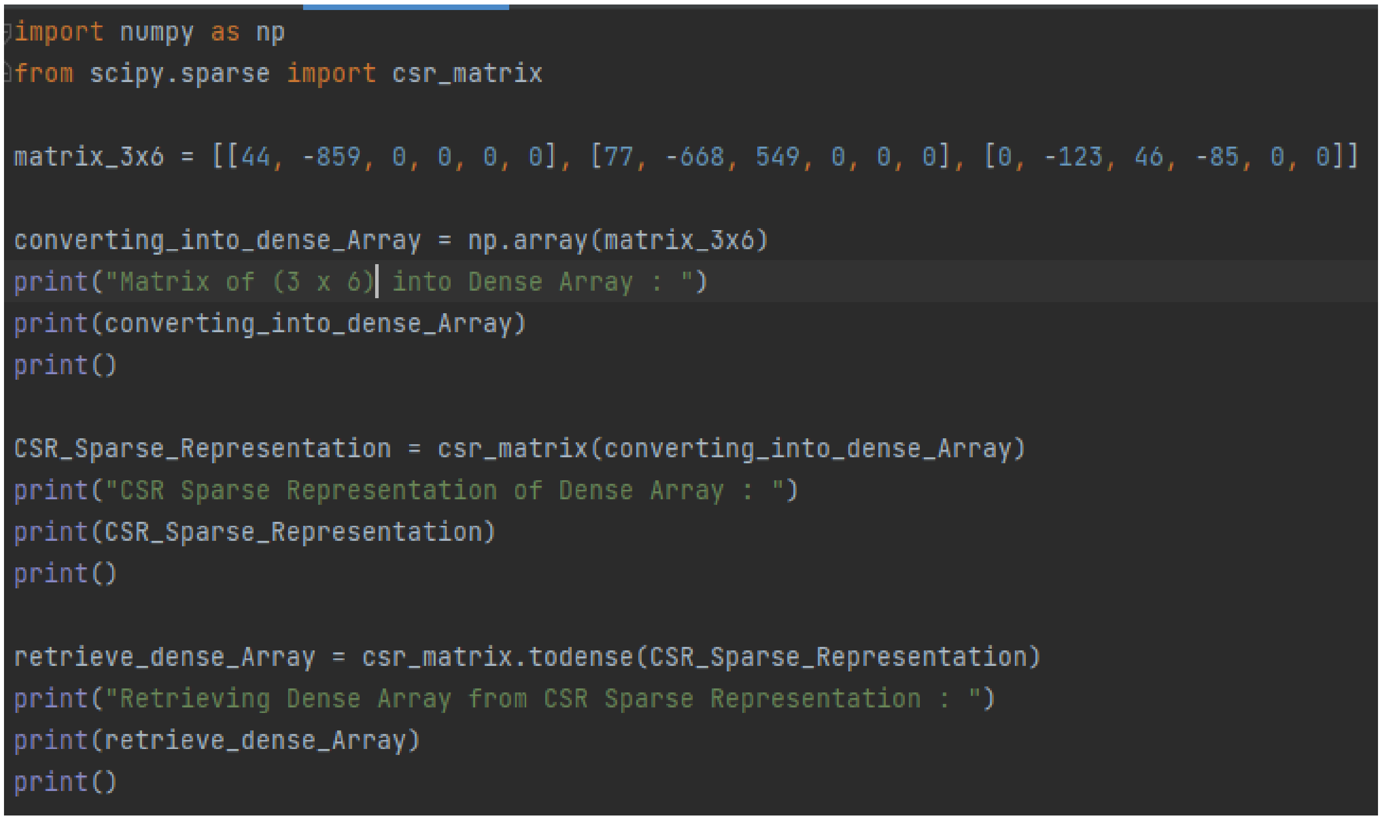
```
Triangular Matrix  :  [[-4, 0, 0, 0], [65, -88, 0, 0], [-24, -41, 59, 0], [-7, -32, -14, -99]]
After Storing Triangular Matrix into Unidimensional :  [-4, 65, -88, -24, -41, 59, -7, -32, -14, -99]
```



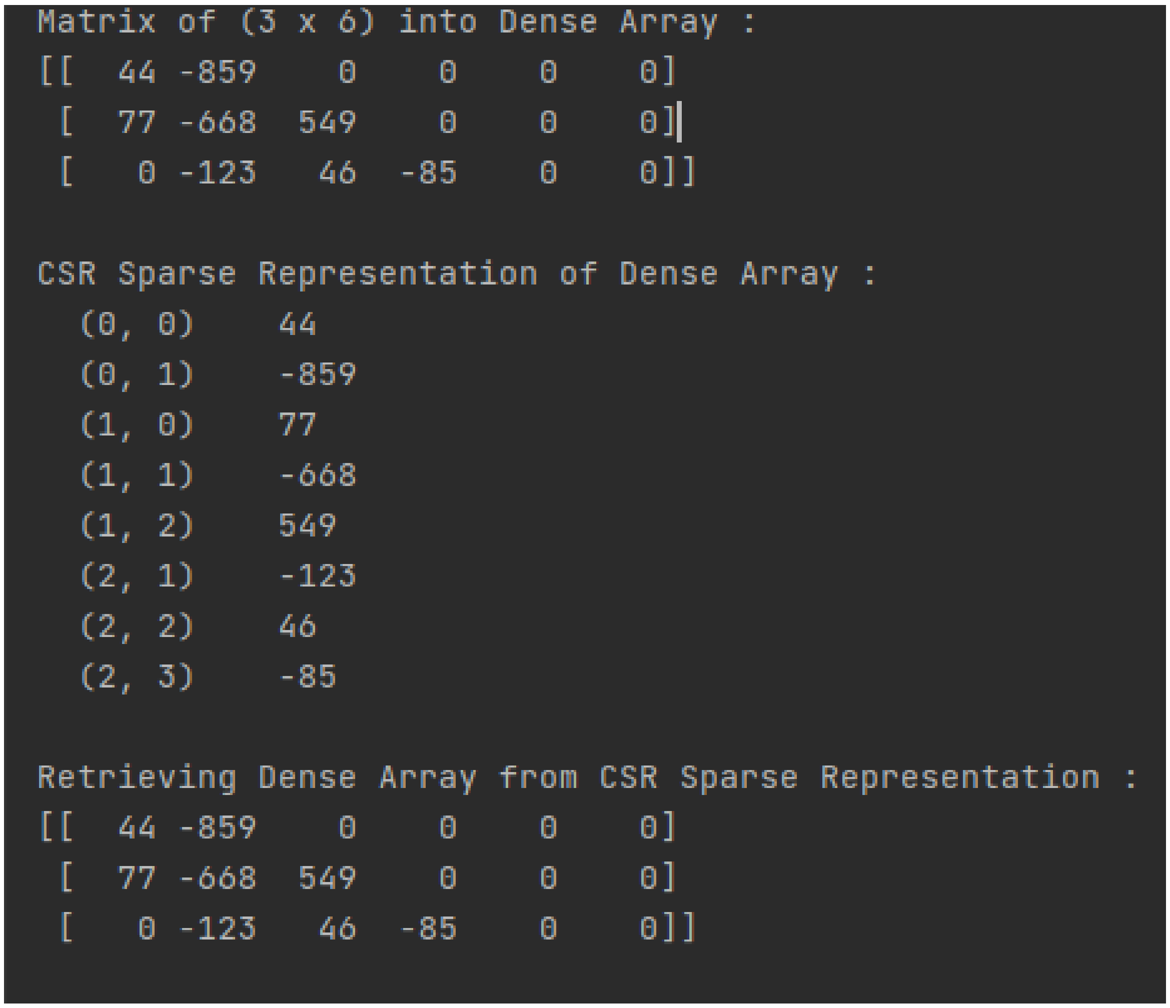
```
Unidimensional array :  [-4, 65, -88, -24, -41, 59, -7, -32, -14, -99]
Retrieving Triangular Matrix from Unidimensional array:  [[-4, 0, 0, 0], [65, -88, 0, 0], [-24, -41, 59, 0], [-7, -32, -14, -99]]
```



```python
def RETRIEVE_TRIANGULAR(U, n):
    A = []

    for i in range(n):
        new_lst = []
        for z in range(n):
            new_lst.append(0)
        A.append(new_lst)

    for j in range(n):
        for k in range(n):
            if k > j:
                A[j][k] = 0
            else:
                A[j][k] = U[int(0.5*j*(j+1)+k)]

    return A


unidimensional_array = [-4, 65, -88, -24, -41, 59, -7, -32, -14, -99]
print("Unidimensional array : ", unidimensional_array)

converted_triangular_matrix = RETRIEVE_TRIANGULAR(unidimensional_array, 4)
print("Retrieving Triangular Matrix from Unidimensional array: ", converted_triangular_matrix)
```



```python
import numpy as np
from scipy.sparse import csr_matrix

matrix_3x6 = [[44, -859, 0, 0, 0, 0], [77, -668, 549, 0, 0, 0], [0, -123, 46, -85, 0, 0]]

converting_into_dense_Array = np.array(matrix_3x6)
print("Matrix of (3 x 6) into Dense Array : ")
print(converting_into_dense_Array)
print()

CSR_Sparse_Representation = csr_matrix(converting_into_dense_Array)
print("CSR Sparse Representation of Dense Array : ")
print(CSR_Sparse_Representation)
print()

retrieve_dense_Array = csr_matrix.todense(CSR_Sparse_Representation)
print("Retrieving Dense Array from CSR Sparse Representation : ")
print(retrieve_dense_Array)
print()
```



```
Matrix of (3 x 6) into Dense Array :
[[  44 -859    0    0    0    0]
 [  77 -668  549    0    0    0]
 [   0 -123   46  -85    0    0]]

CSR Sparse Representation of Dense Array :
  (0, 0)	44
  (0, 1)	-859
  (1, 0)	77
  (1, 1)	-668
  (1, 2)	549
  (2, 1)	-123
  (2, 2)	46
  (2, 3)	-85

Retrieving Dense Array from CSR Sparse Representation :
[[  44 -859    0    0    0    0]
 [  77 -668  549    0    0    0]
 [   0 -123   46  -85    0    0]]
```