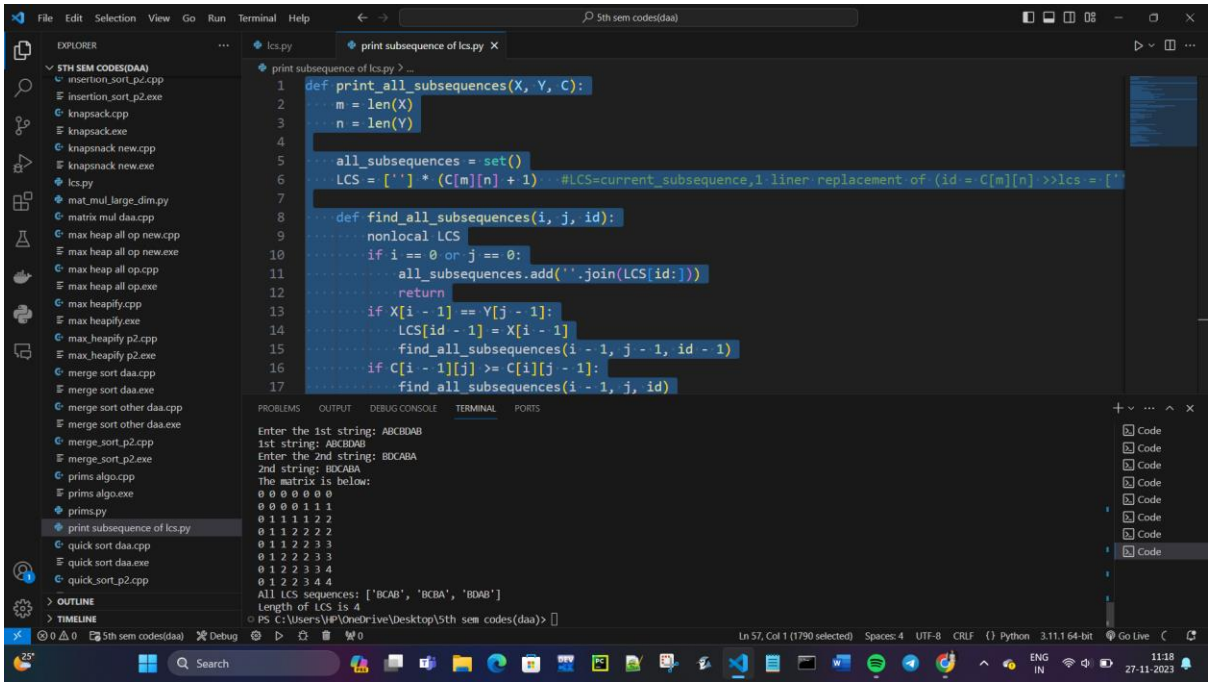
```python
#print all subsequences of lcs
def print_all_subsequences(X, Y, C):
    m = len(X)
    n = len(Y)

    all_subsequences = set()
    LCS = [''] * (C[m][n] + 1)   #LCS=current_subsequence,1 liner replacement of (id = C[m][n] >>lcs = [''] * (id + 1) >>lcs[id] = '\0')

    def find_all_subsequences(i, j, id):
        nonlocal LCS
        if i == 0 or j == 0:
            all_subsequences.add(''.join(LCS[id:]))
            return
        if X[i - 1] == Y[j - 1]:
            LCS[id - 1] = X[i - 1]
            find_all_subsequences(i - 1, j - 1, id - 1)
        if C[i - 1][j] >= C[i][j - 1]:
            find_all_subsequences(i - 1, j, id)
        if C[i][j - 1] >= C[i - 1][j]:
            find_all_subsequences(i, j - 1, id)
#''.join(LCS[id:]): This part joins the elements of the LCS list from index id to the end (: indicates the end of the list) into a single string.

    find_all_subsequences(m, n, C[m][n])

    # Print all subsequences
    filtered_subsequences = [i for i in all_subsequences if len(i) == 4]
    print("All LCS sequences:", filtered_subsequences)

    #print the LCS length
    print("Length of LCS is",C[m][n])

# Main function
X = input("Enter the 1st string: ")
```

```
print("1st string:", X)
Y = input("Enter the 2nd string: ")
print("2nd string:", Y)

m = len(X)
n = len(Y)

C = [[0] * (n + 1) for _ in range(m + 1)]

for i in range(1, m + 1):
    for j in range(1, n + 1):
        if X[i - 1] == Y[j - 1]:
            C[i][j] = C[i - 1][j - 1] + 1
        else:
            C[i][j] = max(C[i - 1][j], C[i][j - 1])

# Print the matrix
print("The matrix is below:")
for i in range(m + 1):
    for j in range(n + 1):
        print(C[i][j], end=" ")
    print()

# Call the function to print all subsequences
print_all_subsequences(X, Y, C)
```


File Edit Selection View Go Run Terminal Help
5th sem codes(daa)
EXPLORER
5TH SEM CODES(DAA)
lcs.py
print subsequence of lcs.py
def print_all_subsequences(X, Y, C):
m = len(X)
n = len(Y)
all_subsequences = set()
LCS = [''] * (C[m][n] + 1) #LCS=current_subsequence,1 liner replacement of (id = C[m][n] >>lcs = ['
def find_all_subsequences(i, j, id):
nonlocal LCS
if i == 0 or j == 0:
all_subsequences.add(''.join(LCS[id:]))
return
if X[i - 1] == Y[j - 1]:
LCS[id - 1] = X[i - 1]
find_all_subsequences(i - 1, j - 1, id - 1)
if C[i - 1][j] >= C[i][j - 1]:
find_all_subsequences(i - 1, j, id)
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS
Enter the 1st string: ABCBDAB
1st string: ABCBDAB
Enter the 2nd string: BDCABA
2nd string: BDCABA
The matrix is below:
0 0 0 0 0 0 0
0 0 0 0 1 1 1
0 1 1 1 1 2 2
0 1 1 2 2 2 2
0 1 1 2 2 3 3
0 1 2 2 2 3 3
0 1 2 2 3 3 4
0 1 2 2 3 4 4
All LCS sequences: ['BCAB', 'BCBA', 'BDAB']
Length of LCS is 4
PS C:\Users\HP\OneDrive\Desktop\5th sem codes(daa)>
Ln 57, Col 1 (1790 selected) Spaces: 4 UTF-8 CRLF Python 3.11.1 64-bit Go Live
Search
ENG IN
11:18 27-11-2023