Practical 5

name:-Virendra A Uplenchwar
Roll no:-A4_B1_05

Code(task1):-

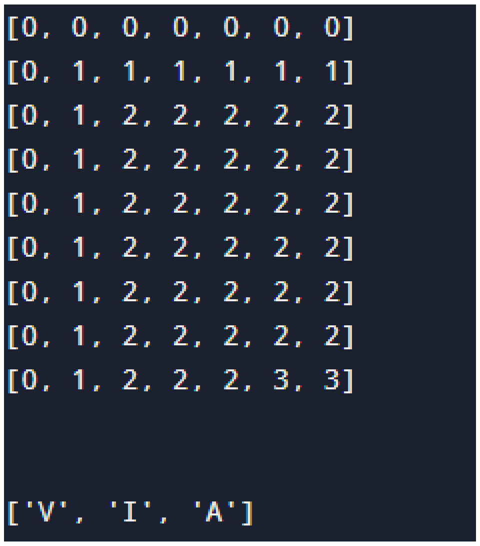
```
string1="VIRENDRA"
string2="VISHAL"
rows,cols=(9,7)
arr = [[0]*cols for _ in range(rows)]
for i in range(1,rows):
    for j in range(1,cols):
        if string1[i-1]==string2[j-1]:
            arr[i][j]=arr[i-1][j-1]+1
        else:
            arr[i][j]=max(arr[i-1][j],arr[i][j-1])

i=rows-1
j=cols-1
chars=[]
while i>0 and j>0:
    if string1[i-1]==string2[j-1]:
        chars.append(string1[i-1])
        i=i-1
        j=j-1
    elif arr[i-1][j]>arr[i][j-1]:
        i=i-1
    else:
        j=j-1

chars.reverse()
string3="".join(chars)

for rows in arr:
    print(rows)
print("\n")
print(chars)
```

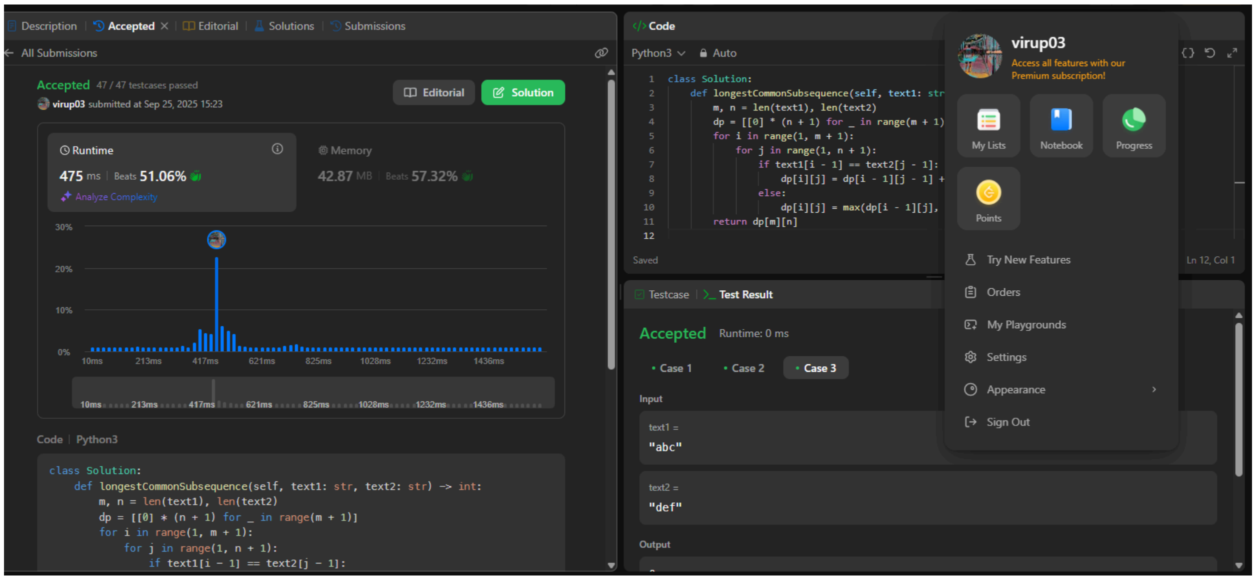
Output:-



```
[0, 0, 0, 0, 0, 0, 0]
[0, 1, 1, 1, 1, 1, 1]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 2, 2]
[0, 1, 2, 2, 2, 3, 3]


['V', 'I', 'A']
```

Leetcode:-



Accepted 47 / 47 testcases passed

virup03 submitted at Sep 25, 2025 15:23

Runtime: 475 ms | Beats 51.06%

Memory: 42.87 MB | Beats 57.32%

Code | Python3

```
class Solution:
    def longestCommonSubsequence(self, text1: str, text2: str) -> int:
        m, n = len(text1), len(text2)
        dp = [[0] * (n + 1) for _ in range(m + 1)]
        for i in range(1, m + 1):
            for j in range(1, n + 1):
                if text1[i - 1] == text2[j - 1]:
```

```
class Solution:
    def longestCommonSubsequence(self, text1: str
        m, n = len(text1), len(text2)
        dp = [[0] * (n + 1) for _ in range(m + 1)
        for i in range(1, m + 1):
            for j in range(1, n + 1):
                if text1[i - 1] == text2[j - 1]:
                    dp[i][j] = dp[i - 1][j - 1] +
                else:
                    dp[i][j] = max(dp[i - 1][j],
        return dp[m][n]
```

virup03

Access all features with our Premium subscription!

Accepted Runtime: 0 ms

Input

text1 =
"abc"

text2 =
"def"

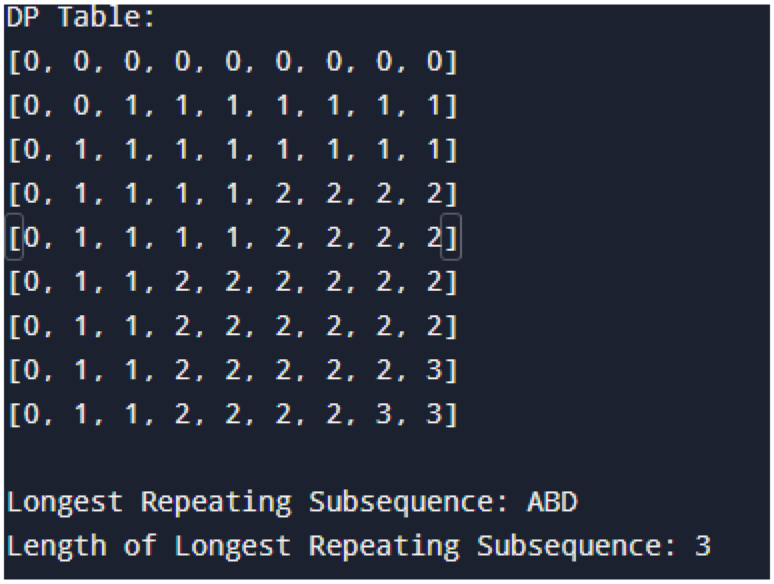
Output

Code(task 2):-

```
def longestRepeatingSubsequence(s1):
    n = len(s1)
    arr = [[0] * (n + 1) for _ in range(n + 1)]
    for i in range(1, n + 1):
        for j in range(1, n + 1):
            if s1[i - 1] == s1[j - 1] and i != j:
                arr[i][j] = arr[i - 1][j - 1] + 1
            else:
                arr[i][j] = max(arr[i - 1][j], arr[i][j - 1])
    i, j = n, n
    chars = []
    while i > 0 and j > 0:
        if s1[i - 1] == s1[j - 1] and i != j:
            chars.append(s1[i - 1])
            i -= 1
            j -= 1
        elif arr[i - 1][j] > arr[i][j - 1]:
            i -= 1
        else:
            j -= 1
    chars.reverse()
    print("DP Table:")
    for row in arr:
        print(row)
    return "".join(chars), arr[n][n]
s1 = "AABEBCDD"
lrs, lrs_length = longestRepeatingSubsequence(s1)

print("\nLongest Repeating Subsequence:", lrs)
print("Length of Longest Repeating Subsequence:", lrs_length)
```

Output:-



```
DP Table:
[0, 0, 0, 0, 0, 0, 0, 0, 0]
[0, 0, 1, 1, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 1, 1, 1, 1]
[0, 1, 1, 1, 1, 2, 2, 2, 2]
[0, 1, 1, 1, 1, 2, 2, 2, 2]
[0, 1, 1, 2, 2, 2, 2, 2, 2]
[0, 1, 1, 2, 2, 2, 2, 2, 2]
[0, 1, 1, 2, 2, 2, 2, 2, 3]
[0, 1, 1, 2, 2, 2, 2, 3, 3]

Longest Repeating Subsequence: ABD
Length of Longest Repeating Subsequence: 3
```