DAA PRACTICAL 5

TASK 1:

Find the similarity between the given X and Y sequence.

X=AGCCCTAAGGGCTACCTAGCTT

Y= GACAGCCTACAAGCGTTAGCTTG

```
//longest common sequence(LCS)


#include <stdio.h>
#include <string.h>


int max(int a, int b)
{
    return a > b ? a : b;
}


v
o
i
d
L
C
S
(
c
h
a
r
*
X
,
c
h
a
r
*
Y
)
```

```
{
int m = strlen(X), n = strlen(Y);

    int dp[m+1][n+1];


    // Fill DP table
    for (int i = 0; i <= m; i++)
        for (int j = 0; j <= n; j++)

if (i == 0 || j == 0)
dp[i][j] = 0;

            else if (X[i-1] == Y[j-1])
                dp[i][j] = dp[i-1][j-1] + 1;
            else
                dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
```

```
//Backtracktogetthe LCS in the form of m,j=n,k=dp[m][n];
    char lcs[k+1];
    lcs[k] = '\0';
```

```
    while (i > 0 && j > 0) {
        if (X[i-1] == Y[j-1]) {
            lcs[--k] = X[i-1];
            i--; j--;
        } else if (dp[i-1][j] > dp[i][j-1])


i
-
-
;

e
l
s
e
            j--;
    }

    printf("LCS Length: %d\n", dp[m][n]);
    printf("LCS: %s\n", lcs);
}

int main() {
    char X[] = "AGCCCTAAGGGCTACCTAGCTT";
    char Y[] = "GACAGCCTACAAGCGTTAGCTTG";
    LCS(X, Y);
    return 0;
}
```

```
[Running] cd "c:\Users\DT USER\Desktop\IA333333\DA
LCS Length: 16
LCS: GCCCTAAGCTTAGCTT

[Done] exited with code=0 in 0.87 seconds
```

TASK-2:

Find the longest repeating subsequence (LRS). Consider it as a variation of the longest common subsequence (LCS) problem.

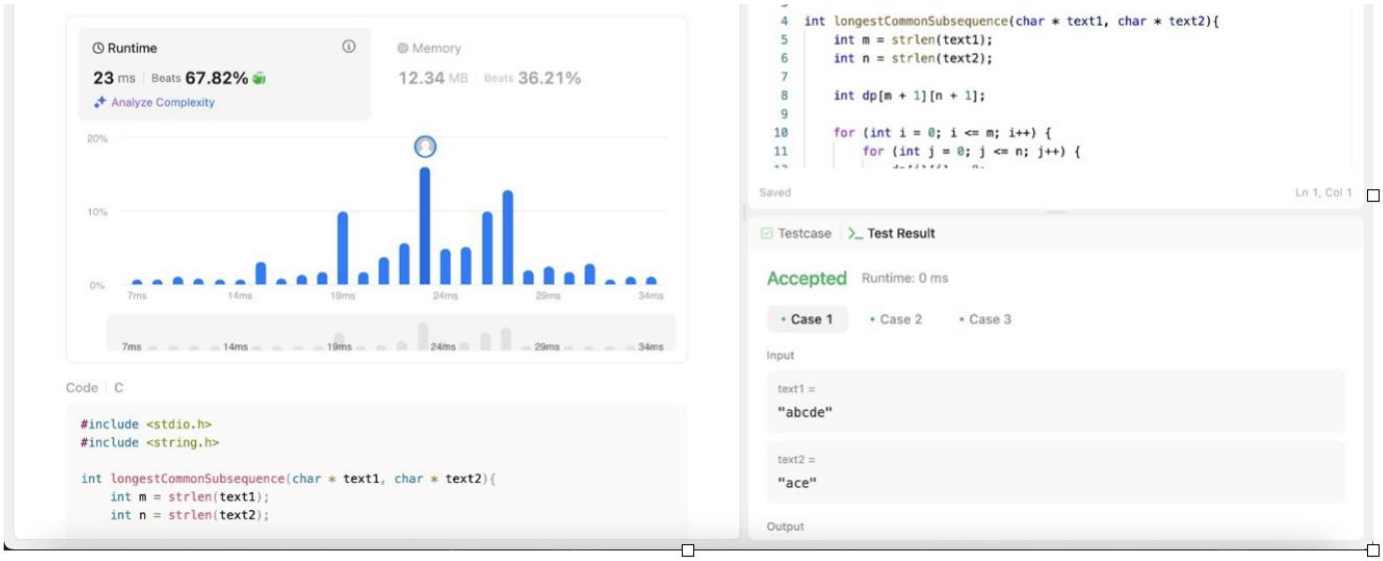
```
// longest repeating sequence(LRS)


#include <stdio.h>
#include <string.h>


int max(int a, int b)
{
     return a > b ? a : b;
}
void LRS(char *str) {

int n = strlen(str);int dp[n+
```

```
1
]
[
n
+
1
]
;

    // Fill DP table
    for (int i = 0; i <= n; i++)
        for (int j = 0; j <= n; j++)
            if (i == 0 || j == 0)
                dp[i][j] = 0;
            else if (str[i-1] == str[j-1] && i
!= j)                 dp[i][j] = dp[i-1][j-1] +
1;
            else
                dp[i][j] = max(dp[i-1][j], dp[i][j-1]);
    // Backtrack to get LRS
    int i = n, j = n, k = dp[n][n];
    char lrs[k+1];


l
r
s
[
k
]
=
'
\
0
'
;
w
h
i
l
e
(
i
>
```

```
0
&
&
j
>
0
)
{
        if (str[i-1] == str[j-1] && i != j) {
            lrs[--k] = str[i-1];
            i--; j--;
        } else if (dp[i-1][j] > dp[i][j-1])
            i--;


e
l
s
e


j
-
-
;
    }
    printf("LRS Length: %d\n", dp[n][n]);
    printf("LRS: %s\n", lrs);
}
int main() {
    char S[] = "AABCBDC";


L
R
S
(
S
)
;
r
e
```

```
t
u
r
n
0
;

}
```

```
[Running] cd "c:\Users\DT USER\Desktop\1A3333
USER\Desktop\1A333333\DAA\A333333\daa 4\daa5\
LRS Length: 3
LRS: ABC

[Done] exited with code=0 in 0.241 seconds
```



LEETCODE: