Returned on _____

Procedure LCS_LENGTH (x, y)

Description - $X = (x_1, x_2, \ldots x_m)$ & $Y = (y_1, y_2 \ldots y_n)$ are two given sequences. The algorithm uses two mxn matrices. C(0:m, 0:n) and B(0:m, 0:n). The matrix B stores the length of lcs & matrix B stores symbol s are computed in row order. This procedure return matrix B & C.

Declaration - global integer c(0:m, 0:n)
char B(0:m, 0:n)
char x(1:m), y(1:n)
local integer m, n, i, j.

Algorithm -

```
m ← LENGTH (x)
n ← LENGTH (y)
for i ← 0 to m do
    c(i,0) ← 0 // fill up 0 in 0 column
repeat
for j ← 0 to n do
    c(0,j) ← 0
repeat
for i ← 1 to m do
    for j ← 1 do n do
        if x(i) = y(j) then,
            c(i,j) ← c(i-1, j-1) + 1
```

Incomplete for :
1) Algorithm
2) Flow Chart
3) Programme Listing
4) Results
5) Comments

```
            B(i,j) ← k
        else
            if c(i-1,j) ≥ c(i,j-1) then
                c(i,j) ← c(i-1,j)
                B(i,j) ← '↑'
            else
                c(i,j) ← c(i,j-1)
                B(i,j) ← '←'
            endif
        endif
    repeat
repeat
return c & B
END_LCS_LENGTH
```

* Non - Recursive

```
Procedure LCS_PRINT I (B,x,m,n)
    global char B(0:m, 0:n)
    global x(x1, x2, x3, ... xm)
    integer m,n,i
    local integer i,j
    STACK S(1:k)

    For i ← m to 1 by -1 do,
        for j ← n to 1 by -1 do,
            if B(i,j) = 'k', then
                PUSH (x(i))
                i ← i-1
            else
                if B(i,j) = '↑' then
                    i ← i-1 and j ← j+1
                endif
            endif
```

Remarks _____ Expt. No. _____ Submitted on _____

Returned on _____

```
        repeat
    repeat
    for i ← top to 1 do
        PRINT (STACK(i))
    repeat
END LCS_PRINT1
```