M03S03 [Recursive Functions and Pointers]: Exercise 01

Problem Description

A recursive function is a function that calls itself. Every recursive function comprises of two essential steps known as base case and recursive step. At some point, the recursive function must encounters a subtask that it can perform without calling itself. This case, where the function does not recur, is called the base case. A recursive method solves a problem by calling a copy of itself to work on a smaller problem. This step is called recursion step.

The Ackermann function A(x,y) is defined for integers x and y by

$$A(x, y) = \begin{cases} y + 1, & \text{if } x = 0 \\ A(x - 1, 1), & \text{if } y = 0 \\ A(x - 1, A(x, y - 1)), & \text{otherwise} \end{cases} \tag{1}$$

Write a program to compute the above shown Ackerman's function using recursion.

Your program must take x and y as input from user during runtime. You have to determine $A(x', y')$, $\forall$ $(x' < x$ or $y' < y)$ exactly once while computing A(x,y).
(Hint: You can use a table to store the values of $A(x', y')$, $\forall$ $(x' < x$ or $y' < y)$.)

Example 1:
$x = 0$ and $y = 0$
Expected Output: 0

Example 2:
$x = 0$ and $y = 5$
Expected Output: 6

Example 3:
$x = 3$ and $y = 0$
Expected Output: 5

Example 4:
$x = 2$ and $y = 3$
Expected Output: 9