

## Exercice

L'algorithme suivant reprend le principe de la fonction `ADD_REC` vue lors du TD2 sauf que cette fois ci, une variable `res` est utilisée pour stocker le résultat de l'appel courant de la fonction.

<pre>ADD_REC(a, b) res ← 0 if a = 0 then:     res ← b else:     res ← ADD_REC(a - 1, b + 1) return res</pre>
--

1. Compléter le tableau ci-dessous pour les valeurs de  $a = 5$  et  $b = 3$ :

<i>a</i>	<i>b</i>	<i>res</i>	Valeur de retour de <code>ADD_REC(a, b)</code>
5	3		

2. Cet algorithme est certainement linéaire en  $a$ . L'opération qui contribue le plus à sa complexité est le nombre de décréments de  $a$ .
- Combien de décréments de  $a$  sont effectuées pour calculer `ADD_REC(5, 3)`?
  - Combien de décréments de  $a$  sont effectuées pour calculer `ADD_REC(5, 3)`?
  - Ces deux appels ont pourtant le même résultat. Modifier l'algorithme pour qu'il soit plus efficace en terme de nombre de décréments de  $a$ .