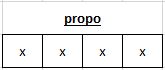
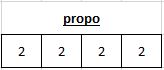
PROPO

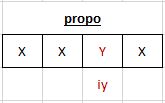


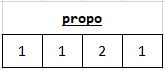
- propoX(**int** x) : remplit ‘propo’ de ‘x’



ex : propoX(2)

- propoXY(**int** x, **int** y, **int** iy) : remplit ‘propo’ de x et un y à la position propo[iy]





ex : propoXY(1, 2, 2)

- propoXouXY(int[] tab) : renvoie un tableau de 2x2 de la forme :



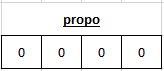
Avec X : chiffre contenu le plus de fois dans tab

Y : chiffre contenu une fois dans tab

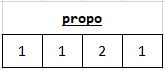
nbX : nombre de X dans tab

iy : emplacement de Y dans tab

ex 1 : propoXouXY(propo)

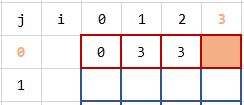


ex 2 : propoXouXY(propo)

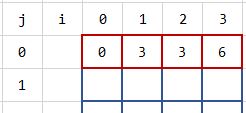


MasterTable

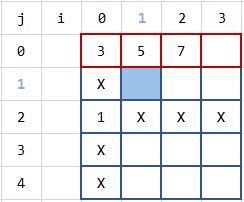
yATrouver() : renvoie true si il reste au moins un « -2 » dans la 1ere ligne de la masterTable



ex 1 : yATrouver() -> true

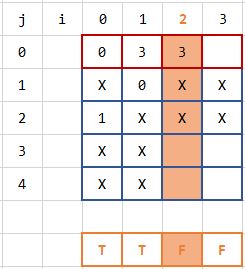
 yATrouver() -> false

jATrouver() : renvoie un couple [iy, jy] correspondant à l’adresse du prochain indice à tester



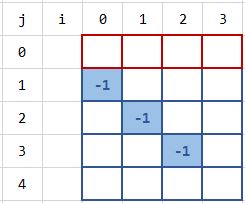
ex : jATrouver() -> [1,1]

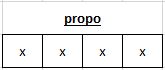
chercheIYMT(int Y) : renvoie la position du 1er Y de la premiere ligne de MT pour lequel colonneTerminee[] est fausse

ex : chercheIYMT(3) -> 2

--------------------------------------------------------------------------

indiceARayer(int iMT, int jMT) : raye (-1) la case MT[jMT][iMT]





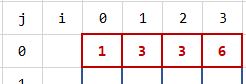
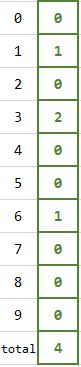
ex : indiceARayer(0,1)

indiceARayer(1,2)

indiceARayer(2,3)

--------------------------------------------------------------------------

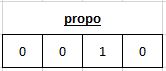
initPremiereLigneMT() : remplit la 1ere ligne de MT d’après IT



--------------------------------------------------------------------------

DIVERS

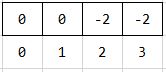
- compteCombien(**int** x, **int**[] tab) : renvoie le nombre de fois que ‘x’ se trouve dans ‘tab’



Ex : compteCombien(0, propo) -> 3

compteCombien(1, propo) -> 1

chercher1erNull(int[] tab) : renvoie l’indice du premier « -2 » trouvé dans tab



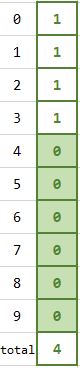
ex : cherche1erNull(tab) -> 2

IT

--------------------------------------------------------------------------

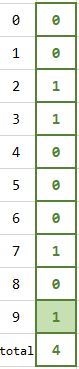
completeIT() : complete IT si possible

ex 1 : completeIT() -> complete avec des zeros



**4 = total**

ex 2 : completeIT() -> complete la derniere case vide



**total – 3 = 1**

**& il reste 1 case à remplir**