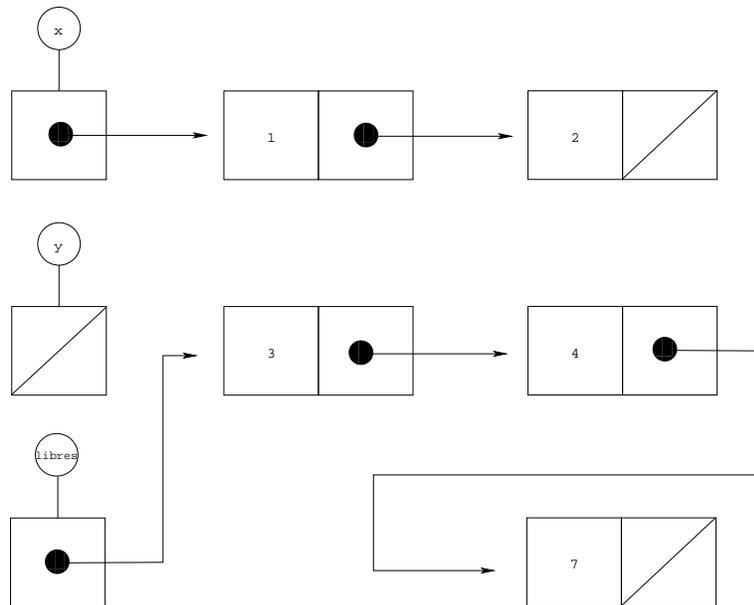


Certains langages, comme Java et Caml, disposent d'un mécanisme automatique de récupération des cellules inaccessibles : le *glaneur de cellules* (GC).

5.6.2 Programmer sans glaneur de cellules

D'autres langages plus anciens, comme C, n'ont pas de glaneur de cellules. La mémoire doit alors être gérée à la main. Cela demande d'ajouter les références r_5 et r_6 à une liste de cellules libres avant d'effectuer l'instruction $y = \text{NULL}$;



Ensuite, à chaque fois que l'on a besoin d'une nouvelle cellule, au lieu de l'allouer systématiquement, on commence par tester si la liste `libres` contient une cellule et si c'est le cas, on en recycle une.

Quand on libère ainsi des cellules, il faut prendre garde à ne pas libérer une cellule qui reste accessible, car elle était partagée. Ainsi, si l'on exécute l'instruction $y = \text{NULL}$; dans l'état $e = [x = r_1, y = r_4]$, $m = [r_1 = r_2, r_2 = \{\text{hd} = 1, \text{tl} = r_3\}]$, $r_3 = \{\text{hd} = 2, \text{tl} = \text{NULL}\}$, $r_4 = r_5$, $r_5 = \{\text{hd} = 3, \text{tl} = r_3\}]$