

Soit un ensemble de données défini comme suit :

```
$donnees = array(
array('lettre' => 'a', 'suivant' => 10),
array('lettre' => 'e', 'suivant' => -1),
array('lettre' => 'e', 'suivant' => 6),
array('lettre' => 'l', 'suivant' => 1),
array('lettre' => 'p', 'suivant' => 8),
array('lettre' => 'o', 'suivant' => 11),
array('lettre' => 'x', 'suivant' => 12),
array('lettre' => 'p', 'suivant' => 3),
array('lettre' => 'r', 'suivant' => 5),
array('lettre' => 'm', 'suivant' => 7),
array('lettre' => 'b', 'suivant' => 3),
array('lettre' => 'b', 'suivant' => 0),
array('lettre' => 'a', 'suivant' => 9)
);
```

Chaque élément de \$donnees est une paire lettre-suivant, où lettre est un caractère à afficher, et suivant est la position de l'élément suivant dans \$donnees (-1 signifiant qu'il n'y a pas d'élément après celui-ci).

Par exemple l'élément 3 correspond à la lettre l qui est suivie par l'élément 1 (c'est à dire un e final)

Le but est d'écrire une fonction acceptant en paramètre un index de départ, et qui permet d'obtenir le mot composé à partir de l'élément à cet index et de tous les éléments suivants. Par exemple l'entrée 2 devrait permettre d'obtenir "exemple", l'entrée 4 "probable", l'entrée 0 "able".

Tester cette fonction avec les entrées 2, 4, 0 et 12.

Exercice librement inspiré d'une énigme du jeu Human Ressource Machine, elle même inspirée du fonctionnement des listes chaînées.