



Exercice 1. Écrire une fonction `pile_vers_liste` qui renvoie la liste des éléments contenus dans une pile.

`pile_vers_liste : 'a t -> 'a list`

Exercice 2. Écrire une fonction `empile` qui vide la première pile pour la transférer sur la seconde.

`empile : 'a t -> 'a t -> unit.`

Exercice 3. (Rotation) Écrire une fonction `tourne` qui effectue la rotation d'une pile. Ainsi, `tourne (a1, a2, a3)` renverra `(a2, a3, a1)`.

`tourne : 'a t -> unit`

Exercice 4. Une pile est constituée d'assiettes colorées et numérotées représentées par le type suivant :

```
type couleur = Bleu | Rouge
and assiette = {c : couleur; n : int };;
```

Écrire une fonction `tri` qui trie la pile d'assiettes en mettant les bleues sur le dessus, les rouges sur le dessous et qui préserve l'ordre de numérotation initial.

`tri : 'a t -> unit`

Exercice 5. Proposer un algorithme de parcours en largeur d'un arbre en utilisant des piles.