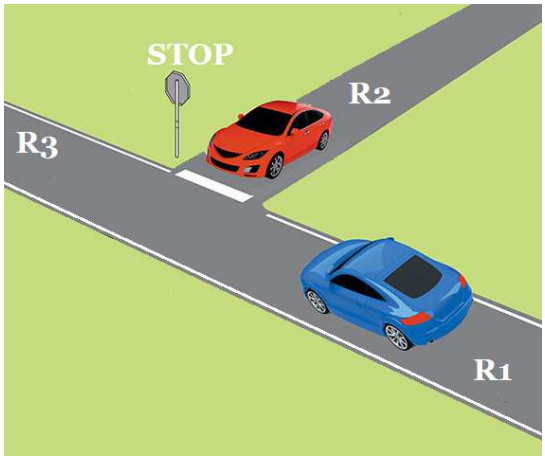
NSI – Structures de données
S'arrêter au stop !



Pour simuler un croisement routier, à sens unique, on utilise 3 files `f1`, `f2` et `f3` représentant respectivement les voitures arrivant sur des routes R1 et R2, et les voitures partant sur la route R3.
La route R2 a un STOP. Les voitures de la file `f2` ne peuvent avancer que s'il n'y a aucune voiture sur la route R1, donc dans la file `f1`.

On souhaite écrire un algorithme qui simule le départ des voitures sur la route R3, modélisée par la file `f3`.
•Dans la file `f1` on représentera la présence d'une voiture par le nombre `1` et l'absence de voiture par `0`.
•Dans la file `f2` on représentera la présence d'une voiture par le nombre `2` et l'absence de voiture par `0`.
• On n'utilisera que les primitives `enfiler(), defiler(), tete(), est_vide()` et `file_vide()`
• On testera l'algorithme sur `f1 : tête <-[0, 1, 1, 0, 1, 0, 1]<- queue`
• On testera l'algorithme sur `f2 : tête <-[0, 0, 2, 2, 2, 2, 2, 0]<- queue`
• Le résultat attendu : `f3 tête <-[0, 1, 1, 2, 1, 2, 1, 2, 2, 2, 0]<- queue`

Que doit faire l'algorithme si les têtes des deux files sont à `0` ?

Que doit faire l'algorithme si la tête de `f1` est à `1` et celle de `f2` à `0` ?

Que doit faire l'algorithme si la tête de `f1` est à `1` et celle de `f2` à `2` ?

Que doit faire l'algorithme si la tête de `f1` est à `0` et celle de `f2` à `2` ?

Que doit faire l'algorithme si l'une des deux files est vide ?

<u>Implémentation en Python de l'algorithme qui modélise ce carrefour :</u>
Ecrire une fonction `croisement(f1,f2)` qui prend en paramètres deux files `f1` et `f2` et qui retourne une file `f3` contenant les voitures (`1` ou `2`) ou l'absence de voiture (`0`) sur la route R3.