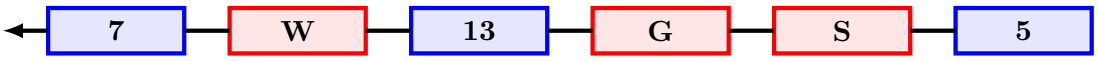
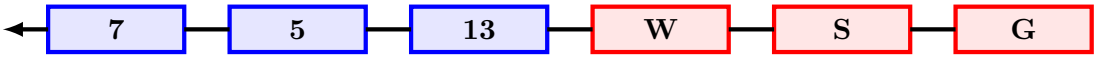
Exercice n° 1 Gare de triage

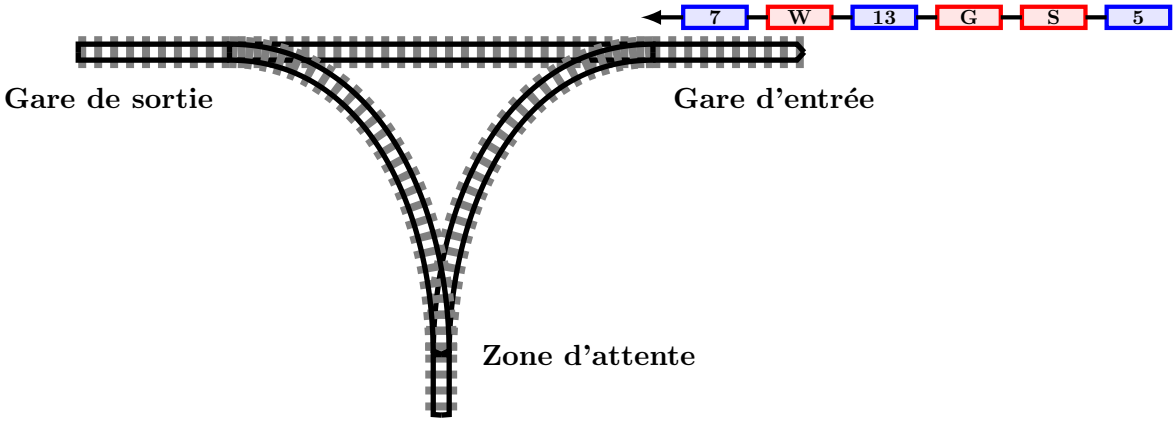
Un train comporte des wagons bleus qui portent un numéro et des wagons rouges qui portent une lettre.



Le chef de gare souhaite séparer les wagons : d'abord tous les bleus et ensuite tous les rouges (l'ordre des wagons bleus n'a pas d'importance, l'ordre des wagons rouges non plus).

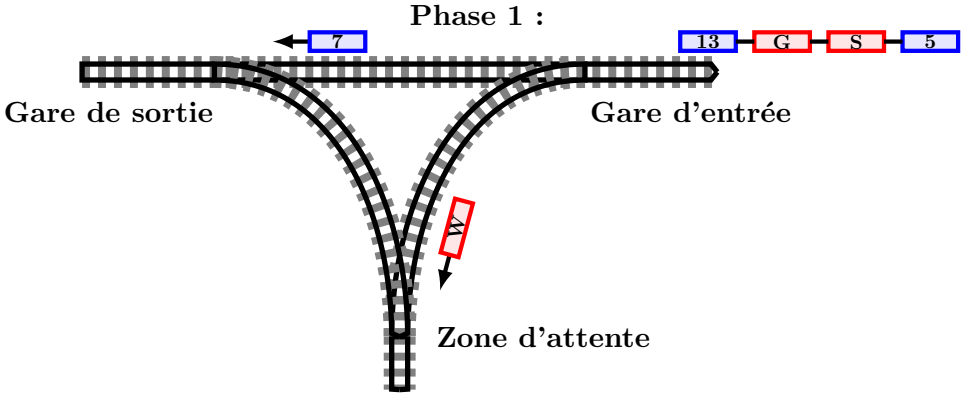


Pour cela, il dispose d'une gare de sortie et d'une zone d'attente : un wagon peut soit être directement envoyé à la gare de sortie, soit être momentanément stocké dans la zone d'attente.

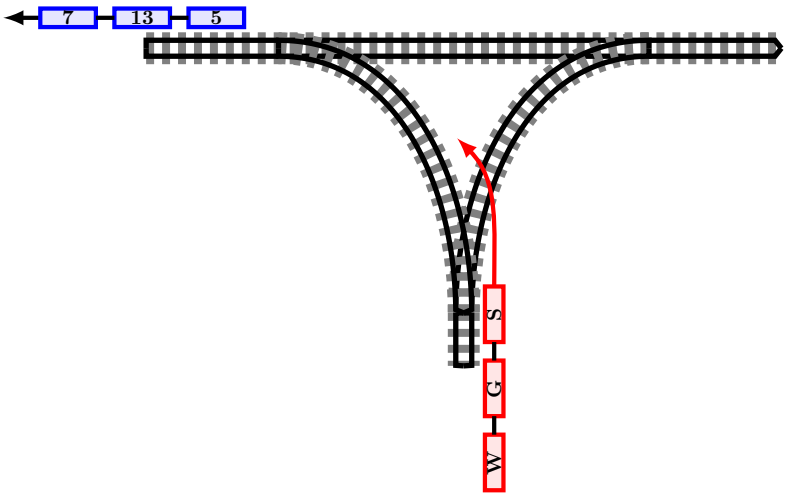


Voici comment procède le chef de gare :

Phase 1 :



Phase 2 :



Aide et instructions :

— Le train est une liste de caractères constitués de nombres (les wagons bleus) et de lettres (les wagons rouges) séparés par des espaces. Par exemple `train = ["G", "6","Z","J","14"]`.
— On teste si un wagon est bleu est regardant s'il est marqué d'un nombre, par la fonction `wagon.isdigit()`.
— le train de départ et le train reconstitué seront traités comme une **FILE**.
— La zone d'attente sera traitée comme une **PILE**. Au départ le train reconstitué et la zone d'attente sont vides.
— Vous trouverez dans le fichier python les fonctions primitives sur les piles et les files.

En suivant les instructions ci-dessus, écrire une fonction `tri_wagons()` qui sépare les wagons bleus et rouges d'un train.

Fonction : `tri_wagons(train)`
Entrée : une chaîne de caractères avec des wagons bleus (nombres) et des wagons rouges (lettres)
Sortie : une chaîne de caractères avec des wagons bleus (nombres) PUIS des wagons rouges (lettres)

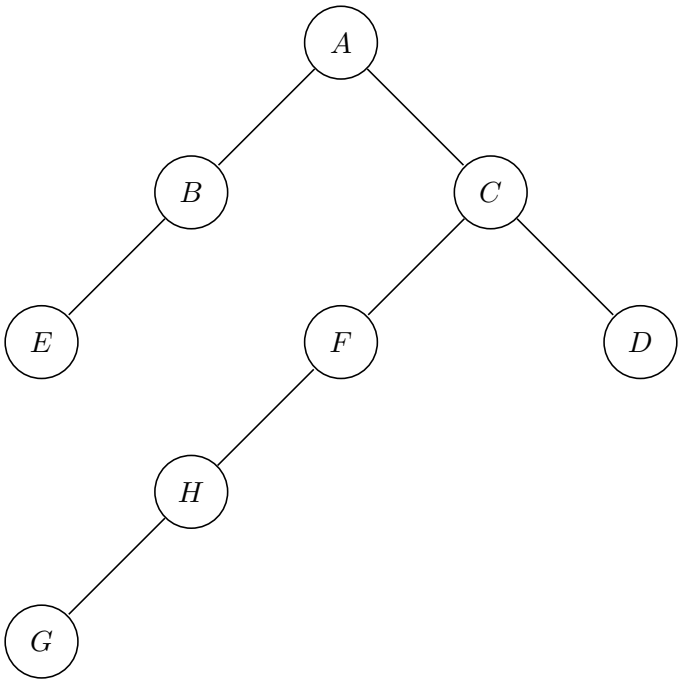
Exemples :

```
tri_wagons(["A", "4", "C", "12"])
>>> ["4", "12", "C" ,"A"]
tri_wagons(["K", "8", "P", "17", "L", "B", "R", "3", "10", "2", "N"])
>>>["8", "17", "3" ,"10" ,"2" ,"N", "R", "B", "L", "P", "K"]
```

Exercice n° 2 Arbre binaire

On donne l'implémentation d'un arbre binaire (voir fichier python)

1. Construire une instance (arb) de la classe **ArbreBinaire** qui a pour représentation :



2. Choisir un type de parcours (préfixe, suffixe, infixe ou largeur) et implémenter le.