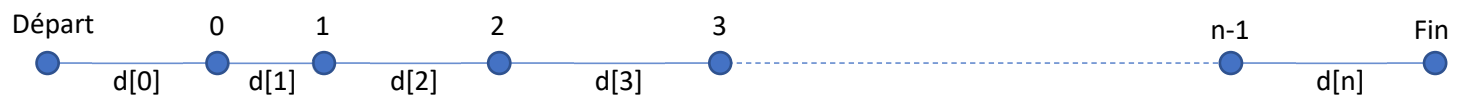


Activité II

Une route comporte n stations-service, numérotées dans l'ordre du parcours, de 0 à $n - 1$.

La première est à une distance $d[0]$ du départ, la deuxième est à une distance $d[1]$ de la première, la troisième à une distance $d[2]$ de la deuxième, etc., la fin de la route est à une distance $d[n]$ de la n -ième (dont le numéro est donc $n-1$) et dernière station-service comme illustré ci-dessous :



Un automobiliste prend le départ de la route avec une voiture dont le réservoir d'essence est plein. Sa voiture est capable de parcourir une distance r (mais pas plus !) avec un plein.

L'objectif est de déterminer dans quelles stations-service l'automobiliste doit s'arrêter pour arriver à destination et en faisant le moins d'arrêts possible.

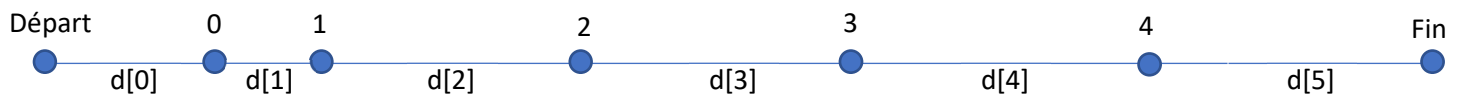
1. Donnez une condition nécessaire et suffisante pour que l'automobiliste puisse effectuer le parcours.
On la supposera réalisée par la suite.

L'automobiliste souhaite arriver à destination. Il n'est pas certain que la jauge de son réservoir à essence fonctionne. (il ne sait pas ce qu'il lui reste)

2. Quelle stratégie lui permet d'être certain de ne pas tomber en panne sèche. Qu'en pensez-vous ?

On suppose dans toute la suite que la jauge fonctionne correctement.

3. Dans quelle situation est-il obligé de s'arrêter à toutes les stations-services ?
Exemple : $r=100$ et $n=5$.



Donnez un exemple de 6 valeurs de distance qui l'obligeront à s'arrêter à chaque station-service.

4. L'automobiliste désire faire le plein le moins souvent possible et arriver à destination. Quelle doit être sa stratégie ?
5. Ecrivez un programme en langage Python qui détermine à quelles stations-service il doit s'arrêter.
 - Le résultat sera donné sous la forme d'une liste indiquant le numéro des stations-service (nombre compris entre 0 et $n-1$).
 - Ce programme devra indiquer si la condition nécessaire et suffisante de la question 1 est vérifiée.
 - En cas d'impossibilité, le programme renverra une liste contenant la chaîne de caractères « Impossible ».
6. Testez votre programme avec 11 stations-services dont la liste des distances est la suivante d : [20 , 40 , 31 , 68 , 45 , 37 , 25 , 106 , 54 , 120 , 86 , 59] et $r = 150$.

Pour aller plus loin : notion de complexité.

https://fr.wikipedia.org/wiki/Analyse_de_la_complexit%C3%A9_des_algorithmes#Complexit%C3%A9_comparatif

7. Votre algorithme doit-il parcourir une liste ou un tableau à deux dimensions (une liste de liste) ? En déduire sa complexité.