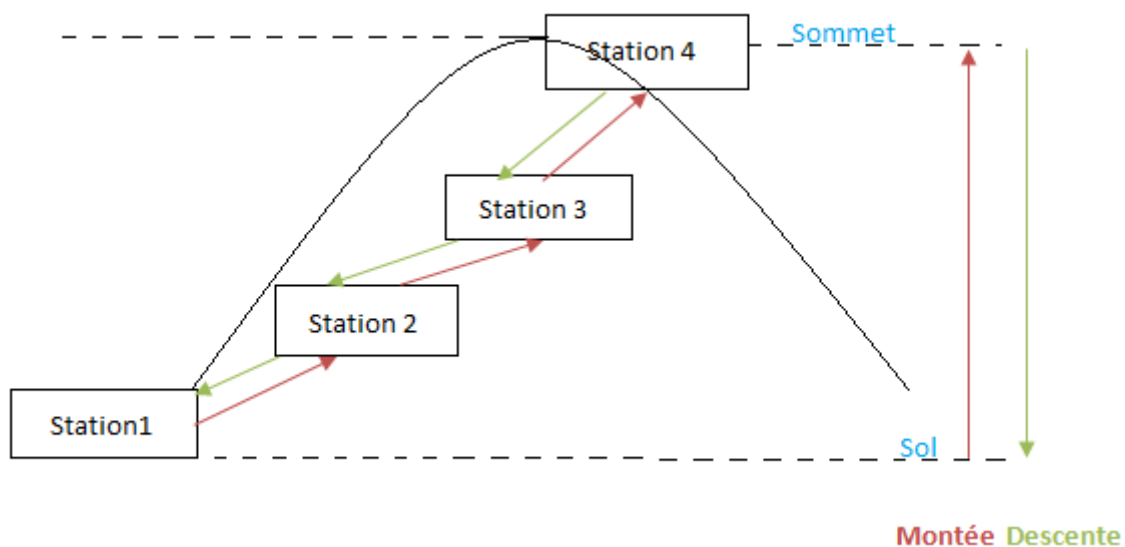


<b>esprit</b> Ecole Supérieure Privée d'Ingénierie et de Technologies	<b>Objectif (6):Les listes doublement chaînées (Partie 1)</b>	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire: 2020-2021

On souhaite développer une application qui permet de gérer une compétition d'escalade sur une montagne. La montagne est composée de différentes stations d'arrêt entre le sol et le sommet. Quatre alpinistes ont participé à ce challenge. L'alpiniste gagnant doit monter et descendre la montagne en un minimum de temps.

Exemple :



Une station d'arrêt est caractérisée par :

- son numéro (le numéro doit être **unique** et supérieur ou égal à 1)
- un tableau de quatre alpinistes

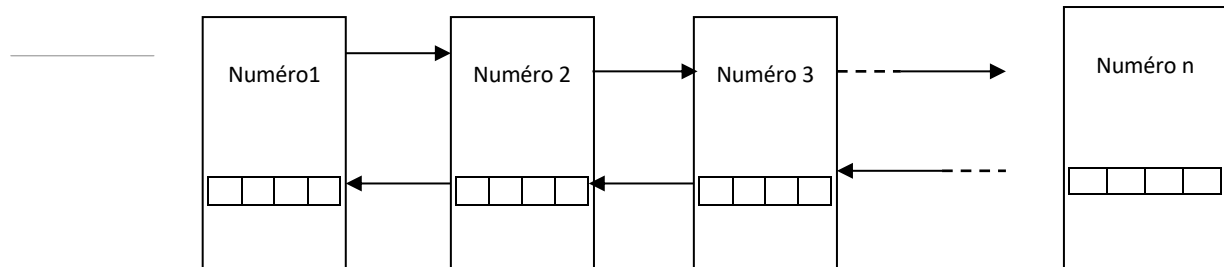
Chaque alpiniste  $i$  (d'indice  $i$  du tableau) est caractérisé par :

- Le temps qu'il a mis pour arriver à la station dans la montée.
- Le temps qu'il a mis pour arriver à la station dans la descente.

On suppose que le chemin bidirectionnel des stations d'arrêt reliant le sol et le sommet de la montagne est représenté par une liste doublement chaînée.

**Objectif (6): Les listes doublement chaînées (Partie 1)**

 Matière : Programmation Procédurale II  
Support : Prosit  
Séance : APP  
Classe(s) : 1A

 Unité pédagogique : Algorithmique &  
Programmation  
Année universitaire: 2020-2021


1. Déclarer les structures nécessaires.

Avant de commencer la compétition, les organisateurs doivent créer le chemin de stations à suivre par les alpinistes. Pour ce faire, il faut initialiser la liste représentant le chemin de l'escalade et y ajouter les stations une par une.

2. Ecrire la fonction **init\_chemin** qui permet d'initialiser la liste représentant le chemin de l'escalade à une liste vide.
3. Ecrire la fonction **saisir\_station** qui permet de saisir une station tout en initialisant :
  - Le temps à la montée à 0.
  - Le temps à la descente à 0.
4. Ecrire la fonction **chercher\_station** qui permet de vérifier l'existence d'une station donnée par son numéro et de retourner un pointeur sur la cellule contenant cette station.
5. Ecrire la fonction **ajouter\_station** qui permet d'insérer une station dans le chemin de l'escalade.  
NB. L'insertion des stations se fera en ordre croissant selon le numéro.
6. Ecrire la fonction **afficher\_chemin** qui permet d'afficher la liste représentant le chemin des stations.
7. Ecrire la fonction **MAJ\_Station** qui permet d'actualiser le temps mis par un alpiniste i en montée ou en descente au niveau d'une station. (Faire les contrôles de saisie nécessaires)
8. Ecrire la fonction **Score\_Alpiniste** qui permet de retourner le score d'un alpiniste donnée par son indice i.

<b>esprit</b> Ecole Supérieure Privée d'Ingénierie et de Technologies	<b>Objectif (6):Les listes doublement chaînées (Partie 1)</b>	
	Matière : Programmation Procédurale II Support : Prosit Séance : APP Classe(s) : 1A	Unité pédagogique : Algorithmique & Programmation Année universitaire: 2020-2021

9. Ecrire la fonction principale main qui permet d'initialiser le chemin de la compétition représenté par la liste doublement chaînée et qui, à travers le menu suivant permet de faire les appels des fonctions nécessaires.

#####MENU#####

1. Ajouter une station
3. Afficher le chemin de la compétition
4. Mettre à jour tous les stations
5. Afficher les scores des trois alpinistes
0. Quitter

#####