DEVOIR DE CONTROLE DU SEMESTRE 2

Matière : *INFORMATIQUE*

Filières : *MP1 – PT1 – PC1 – BG1* ***Durée :*** *1 h*

Exercice 1

On dit qu'une liste `L` à `n` éléments entiers possède un "*pic*" à la position `i` si la valeur `L[i]` est strictement plus grande que `L[i-1]` et `L[i+1]`.

N.B : une liste `L` ne peut pas avoir de pic en position `0`, ni en position `n-1`. De plus, une liste à moins de trois éléments ne peut pas posséder de pics.

Écrire une fonction **`nombrePics(L,n)`** qui calcule et renvoie le nombre de pics d'une liste `L` à `n` entiers.

Exercice 2

Lors d'une course à vélo composée de `n` étapes, on s'intéresse aux altitudes des différentes lignes d'arrivée et de la ligne de départ de la course. Ces altitudes sont mémorisées dans une liste `A` tels que : `A[0]` représente l'altitude du point de départ et `A[1]` l'altitude du point d'arrivée de la première étape, `A[2]` représente l'altitude du point d'arrivée de la deuxième étape et ainsi de suite jusqu'à `A[n]` qui est l'altitude du point d'arrivée de la dernière étape.

Chaque étape est soit "*montante*" soit "*descendante*" et on appelle *dénivelé* l'écart d'altitude entre deux étapes (un dénivelé est toujours positif, que l'étape soit montante ou descendante).

1. Écrire une fonction **`nombreEtapesMontantes`** qui prend en paramètre une liste d'altitudes, calcule et retourne le nombre d'étapes "montantes".
2. Écrire une fonction **`sommeDeniveles`** qui prend en paramètre une liste d'altitudes, calcule et retourne la somme de tous les dénivelés.

Exemple :

Pour la liste **`A = [500, 1100, 1200, 800, 200, 300]`**, le résultat de l'appel à **`nombreEtapesMontantes`** est **3** (il y a trois étapes montantes : les deux premières et la dernière) et la fonction **`sommeDeniveles`** retourne **`1800`** (600 + 100 + 400 + 600 + 100).

Exercice3

On dit qu'une chaîne de caractères *c* est un carré s'il existe une chaîne de caractères *u* telle que *c=uu*. Par exemple, 'papa' est un carré mais 'maman' n'est pas un carré.

1. Écrire une fonction `est_Carre(c)` qui prend en argument une chaîne de caractères *c* et qui retourne le booléen *True* si *c* est un carré, et *False* sinon.

On dit qu'une chaîne de caractères *s* contient un facteur carré, s'il existe un carré *non vide c* et deux chaînes *u* et *v* (éventuellement vides) telles que *s=ucv*. Par exemple, 'maman' contient un facteur carré puisque *u*=' ' , *c*='mama' et *v*='n' conviennent.

2. Écrire une fonction `contient_carre(s)` qui prend en argument une chaîne de caractères *s* et qui retourne le booléen *True* si *s* contient un carré, et *False* sinon.