Questions :

1. De façon global, il faut prendre un fichier, effectuer les opérations nécessaires puis afficher le résultat dans un autre fichier ?
2. Si on ne prend pas toutes les cargaisons dans le premier bâtiment où se trouve le camion, on continue d’en prendre là où on continue aux autres bâtiments ? (prendre le min entre les deux)

Inputs :

* Nombre total de boîtes à transporter
* Capacité maximale d’un camion
* Positions des bâtiments d’entreposage impliqués dans la cargaison courante
* Nombre de boîtes disponibles à chaque point de cargaison

Étapes :

1. Trouver la position du camion :
   1. Récupérer les données dans le fichier texte,
   2. Les mettre dans des Listes chainées,
      1. [Nombre total de boite, capacitée max du camion] (1)
      2. [Nombre de boite dans chaque bâtiment] (2)
      3. [Les bâtiments] (3)
   3. Trouver la valeur max dans (2), retourner son index i. La position du camion correspond au couple (2i, 2i+1) (a, b).
2. Calculer toutes les distances entre la position du camion et les autres bâtiments.
   1. Implémenter la formule de haversine (avec quatre paramètres : , )
   2. Utiliser la formule en itérant sur les éléments de (3) deux par deux. Par exemple haversine (a, (3)[0], b, (3)[1]), haversine (a, (3)[2], b, (3)[3]), haversine (a, (3)[4], b, (3)[5]),… en rajoutant les distances dans une nouvelle liste chainée (4) au fur et à mesure.
3. Effectuer les modifications au niveau des boites déjà chargées.
   1. Trouver la plus petite distance (index i)
   2. La retirer de la liste (4) puis la rajouter dans une nouvelle liste (itérer entre a et b jusqu’à obtenir une liste ordonnée des distances (4’), des coordonnées (3’) et des boites dans chaque bâtiment (2’). Donc, trois nouvelles listes)
   3. Appliquer Algo1
4. Afficher (2)’, (3’) et (4)’ dans args [1].

Algo1(need update)

nbBoiteTransportée 0

nbBoiteRestant Nb de boite à transporter

espace restant capacité max du camion

i = 0

while nbBoiteRestant != 0 :

If (2’)[i] < espace restant :

If nbBoiteTransportée < Nb de boite à transporter :

nbBoiteTransportée += (2’)[i]

nbBoiteRestant -= (2’)[i]

espace restant -= (2’)[i]

(2’)[i] = 0

I++

else :

nbBoiteRestant = 0

else if (2’)[i] < capacité max du camion :

nbBoiteTransportée += nbBoiteRestant

(2’)[i] -= nbBoiteRestant

espace restant = 0

nbBoiteRestant = 0

else :

(2’)[i] -= Nb de boite à transporter

nbBoiteTransportée = Nb de boite à transporter

nbBoiteRestant = 0

Algo2

i = 0

While Nb de boite à transporter != 0 :

If (2’)[i] <= capacité max du camion {

if Nb de boite à transporter >= capacité max du camion

Nb de boite à transporter -= (2’) [i]

var = (2’)[i] - capacité max du camion

If var <= 0 :

(2’)[i] = 0

I++

else :

(2’)[i] = var

I -=0

else:

Nb de boite à transporter = 0

(2’)[i] -= Nb de boite à transporter

else :

if Nb de boite à transporter >= capacité max du camion

Nb de boite à transporter -= capacité max du camion

(2’)[i] -= capacité max du camion

If (2’)[i] = 0:

i++

else :

I -=0

else:

Nb de boite à transporter = 0

(2’)[i] -= Nb de boite à transporter