

1. Python

Dans ces travaux pratiques, vous allez apprendre l'algorithmie en python.

1.1. Tableaux avancés

Q : Dans un programme [hamming.py](#) et à l'aide d'une double iteration (sur 2 listes en même temps), calculer la distance de Hamming sur 2 chaines de caractères :

Pré-requis : les 2 chaînes doivent faire la même longueur.

La distance de Hamming vaut 0 si les deux chaînes sont identiques. À chaque fois qu'une lettre est différente, la distance est incrémenté de 1, par exemple

toto & papa = 4

toto & moto = 1

lythium & pentium = 4

Q : écrire un programme [pile_LIFO.py](#) pour manipuler une pile (LIFO : Last In First Out) :

On sortira du programme quand l'utilisateur saisira exit

imaginer une pile d'assiette, la pile est au départ vide, la pile ne peut faire que 10 assiettes de haut, on ne peut ajouter une assiette que sur le dessus, on ne peut retirer que l'assiette du dessus

si l'utilisateur choisit 'afficher' ou 'a', le programme affiche la pile, l'élément d'en haut en premier

si l'utilisateur choisit 'taille' ou 't', le programme affiche le nombre d'assiette

si l'utilisateur choisit 'empiler' ou 'e', le programme propose à l'utilisateur de rentrer une valeur et il l'ajoute à la pile, si la pile fait déjà 10 assiettes, le programme dit à l'utilisateur « buffer overflow »

si l'utilisateur choisit 'retirer' ou 'r', le programme enlève l'assiette et affiche la valeur de l'assiette, si la pile est vide, le programme affiche « buffer underflow »

Q : écrire un programme [queue_FIFO.py](#) pour manipuler une file ou une queue (FIFO First In First Out) :

On sortira du programme quand l'utilisateur saisira exit

imaginer une caisse de supermarché, la queue est au départ vide, la queue ne peut faire que 10 personne en attente, on ne peut ajouter une personne qu'à la queue (la dernière), on ne peut retirer une personne qu'au début (la première)

si l'utilisateur choisit 'afficher' ou 'a', le programme affiche la file, la personne en début de file élément en premier

si l'utilisateur choisit 'taille' ou 't', le programme affiche le nombre de personne dans la file

si l'utilisateur choisit 'enqueue' ou 'e', le programme propose à l'utilisateur de rentrer une valeur et il l'ajoute à la fin de la queue, si la queue fait déjà 10 personnes, le programme dit à l'utilisateur « ajouter nouvelle caissière »

si l'utilisateur choisit 'retirer' ou 'r', le programme enlève la première personne et affiche la valeur

de la première personne, si la file est vide, le programme affiche « va faire de la pub pour avoir plus de client »

Q : Buffer tournant : écrire un programme `buffer_tournant.py` qui :

On sortira du programme quand l'utilisateur saisira exit,

soit un tableau contenant 5 valeurs. [-2, -1, 0, +1, +2]

Le but est de faire un tourniquet, le programme se met sur 0 et affiche 0

[-2, -1, 0, +1, +2]

Si l'utilisateur tape 'gauche' ou 'g', le tourniquet se déplace sur la gauche, affiche -1

[-2, -1, 0, +1, +2]

Si l'utilisateur tape encore 'g', le tourniquet se déplace à gauche et affiche -2

[-2, -1, 0, +1, +2]

Si l'utilisateur tape encore 'g', le tourniquet se déplace à gauche, comme il n'y a rien, il passe à +2

[-2, -1, 0, +1, +2]

Si l'utilisateur tape encore 'droite' ou 'd', le tourniquet se déplace à droite et affiche -2

[-2, -1, 0, +1, +2]