

Exercice N° 1 :

On donne un nombre réel X et un nombre entier n positif. On désire calculer la somme suivante:

$$S = \sum_{i=0}^n \frac{X^i}{i!}$$

1. Ecrire une fonction **def Empiler_Fat(n, P1)** permettant d'empiler tous les appels de la fonction factorielle jusqu'à n dans une pile $P1$. La fonction retourne $P1$
2. Ecrire une fonction **def Empiler_Puiss(n,x,P2)** permettant d'empiler tous les appels de la fonction puiss jusqu'à n dans une pile $P2$. La fonction retourne $P2$.
3. Ecrire une fonction **def Empiler_Frac(P1 , P2)** qui permet de construire la pile $P3$ à partir de la pile $P1$ et la pile $P2$ en empilant la fraction $X^i/i!$
4. Ecrire une fonction **def Somme(n ,x)** qui retourne la somme S .

Exercice N° 2 : (évaluation des expressions arithmétiques)

La notation polonaise inversée est une façon d'écrire les expressions arithmétiques sans indiquer de multiples parenthèses. Une expression arithmétique écrite en notation polonaise inversée est une suite de nombres et d'opérateurs (+, -, / ou *) et se lit (et s'évalue) de gauche à droite de la façon suivante :

- lorsqu'un nombre est lu, il est empilé sur une pile (initialement vide)
- lorsqu'un opérateur est lu, les deux premiers éléments de la pile sont récupérés et dépilés, l'opération est effectuée sur les deux valeurs obtenues, et le résultat est empilé ;
- Le résultat de l'expression est le premier (et normalement le seul) élément de la pile à la fin de la lecture.

Question 1. Appliquer la méthode exposée pour calculer : $17 \ 10 \ -$, $3 \ 28 \ 7 \ / \ +$ et $3 \ 28 \ + \ 7 \ /$. Comment s'écrivent ces expressions en notation classique ?

Question 2. Traduire en notation polonaise inversée les expressions suivantes :

$(19 * 6) - 7$ et $(4 - 2 * (7 + 6)) + 3 * 5$

Question 3. Ecrire les primitives nécessaires de gestion d'une pile sous forme d'une liste: `PileVide`, `EstVide`, `Empiler`, `Depiler`, et `SommetPile`

Question 4. Écrire une fonction d'entête: **def EstChiffre(c)** qui renvoie 1 si le caractère passé en paramètre est un chiffre et 0 sinon. (Rappeler bien que les chiffres 0,1,...9 sont aussi des caractères)

Par exemple : **EstChiffre('+')=0** et **EstChiffre('4')=1**

Question 5. Écrire une fonction d'entête: **def Convertir(c)** qui permet de convertir un caractère qui représente un chiffre en entier.

Par exemple : **Convertir('2')=2**; et **Convertir('0')=0**

Question 6. Écrire une fonction d'entête: **def Evaluer(expression)** qui évalue une expression polonaise inversée, composée d'entiers entre 0 et 9 et des quatre opérations élémentaires

Question 7. Écrire une fonction d'entête: **def EvaluerTexte(Fsrc, Fdest)** qui permet d'évaluer le contenu du fichier source `src` ensuite stocker le résultat de chaque calcul dans le fichier destination `dest`)

Exercice N° 3 :

On désire maintenant introduire une notion de priorité dans la file d'attente. La priorité est codé par un nombre entier positif, 0 étant le moins prioritaire. Lorsque l'on retire un élément de la queue, on veut obtenir le plus ancien avec la priorité la plus haute. Lors du dépôt d'un élément on indique sa priorité:

def deposer_dans_queue(file, element, priorite);

1-Définir la structure de données adéquate pour gérer la file avec priorités.

2-Ecrire les fonctions de dépôt et de retrait

Exercice N° 4 :

Dans une gare, un guichet est ouvert. Les clients arrivent à des dates aléatoires et rentrent dans une queue. L'intervalle entre l'arrivée de deux clients successifs est un nombre aléatoire entre 0 et **INTERVALLE_MAX** (les dates sont des entiers indiquant des secondes). Lorsque le guichetier a fini de traiter un client, il appelle le client suivant dont le traitement va avoir une durée aléatoire entre 0 et **DUREE_TRAITEMENT_MAX**.

a) Définir les structures de données pour l'algorithme de simulation.

b) Écrire une fonction CreerListeClients, qui crée une file de clients, le nombre de clients étant saisi au clavier. Cette fonction initialise aussi la date d'arrivée et la durée d'attente de chacun des clients. On supposera que le premier client est arrivé à 8h.

c)Écrire une fonction d'affichage qui affiche le numéro de chacun des clients, sa date d'arrivée et sa date de fin de traitement en format (h min sec).

Exercice N° 5 :

1. Importer

1. Téléchargez le fichier csv suivant : [utilisateurs](#). Créer un script python dans votre dossier de travail et déplacez le fichier csv précédent dans le même dossier.
2. Créer une fonction Python permettant d'importer le fichier dans une table. La table renvoyée est une liste de dictionnaires Python.
3. Utiliser cette fonction pour créer la table utilisateurs.

2. Sélectionner

Pour toutes les sélections à effectuer, on partira de la table crée à la question précédente.

On donnera une fonction Python par question (pouvant appeler d'autres fonctions).

1. Créer la table des enregistrements des joueurs ayant moins de 300 points au premier jeu.
2. Créer la table des enregistrements des filles.
3. Créer la table des enregistrements des utilisateurs dont l'adresse mail fait partie du domaine @example.fr

3. Projecter

Pour chaque *projection* on créera une fonction qui retourne une table.

Celle-ci sera toujours une liste de dictionnaires, comme précédemment.

1. *Meilleurs scores*
 - a. Créer les projections des meilleurs scores pour chaque jeu.
 - b. Utiliser cette fonction pour créer la liste des meilleurs scores pour chaque jeu, triées par score croissant.
 - c. Calculer le score moyen de chaque jeu.
2. *Adresses mail.*
 - a. Créer la projection des adresses mail. Le résultat attendu est la liste de toutes les adresses des joueurs.
 - b. Recommencer en ne conservant que les 10 joueurs ayant le plus mauvais score pour chaque jeu.
 - c. Enfin, effacer les doublons.