

TP : Applications des structures de pile et file

Exercice 1: Gestion du parenthésage

Le but de cet exercice est de comprendre comme un éditeur de code (Sublimtext, VS code, Notepad++,...) peut vérifier si un code source n'a pas de problème de parenthésage.

Partie A : Compréhension de l'algorithme

1) Parmi ces textes, expliquer en quoi ils ne sont pas correctement parenthésés.

- a) [[Jean]] [Michel] [[Crochet]] b) (Pierre (Marie (Parenthèse))) () (()
- c) {Anne} { Sophie } { { { Accolades { } } } } d)]) } James Bordure { ([

2) En reprenant les textes suivants, créer les piles des textes suivants en retirant les lettres.

a)											
b)											
c)											
d)											

3) Entourer en rouge les couples de symbole qui se correspondent. Si on les retire des piles cela permet-il de vérifier le bon parenthésage de l'expression.

4) On présente ci dessous un bon parenthésage. En partant de la fin retirer au fur et à mesure 1 couple de parenthèses qui se correspondent. On entourera en rouge le couple que l'on retire puis sur la ligne suivante on écrira les parenthèses restantes.

((())	())	()

5) Reprendre les exemples de la question 1) et faire le même travail. Que remarque-t-on ?

6) Pour vérifier le bon parenthésage de l'expression, on peut utiliser une structure de « pile » qui ajoute les symboles ouvrants quand on parcourt le texte. Proposer un résultat lorsque l'on rencontre un symbole fermant dans les cas suivants :

- a) la pile est vide
- b) le symbole du dessus de la pile lui correspond
- c) le symbole du dessus de la pile ne lui correspond pas

Partie B : Implémentation de l'algorithme

1) Recopier et compléter ce code qui permet de tester le bon parenthésage d'un texte.

Code : parenthesage.py	Code : test_parenthesage.py
<pre>from MaPile import * # On importe la classe Pile avec le fichier MaPile def bonneParenthese(monTexte): mesSymboles = Pile(len(monTexte)) for symbole in monTexte: if symbole in "([{": mesSymboles.empiler(symbole) if symbole in ")]}": # partie du code à compléter pass return mesSymboles.estVide()</pre>	<pre># effectuer les testes sur des expressions txt1 = "((voici) un bon) [parenthésage]" rep1=bonneParenthese(txt1) print(rep1) txt2="((voici) un mauvais{) []parenthésage]" rep2 = bonneParenthese(txt2) print(rep2)</pre>

2) Donner son prototype.

3) Exécuter les testes de [test_parenthesage.py](#)

Exercice 2: Bouton avancer/reculer

Le but de cet exercice est de comprendre comment les boutons « reculer » et « avancer » sont gérés dans un navigateur web (Firefox, Chrome, IE, ...) ou les boutons « annuler » et « rétablir » pour un éditeur de texte.

Partie A : Historique des visites.

Dans cette partie, on utilisera des lettres capitales pour désigner le nom d'un site que l'on visite.

A : Google

B : Mozilla

C : Youtube

D : Wikipédia

E : Amazon

F : SnCF

Pour connaître les sites visités, on construit une pile historique.

- 1) Donner l'ordre des sites visités pour la valeur d'historique suivante : **hist** = [A,F,E,D,A,C,B,A]
- 2) A partir de cette pile d'historique (**hist**), donner la pile qui permet de reculer jusqu'à Wikipédia en passant par les sites précédents dans l'ordre inverse. Attention la pile **hist** augmente en taille à chaque fois.
- 3) Quel est le problème de cette implémentation si on utilise une structure de pile.
- 4) Que se passe-t-il dans **hist** si je retire les sites jusqu'à Wikipédia en les empilant dans une pile temporaire **tmp**. Compléter l'état quand on recule jusqu'à Wikipédia.

hist	A	F	E	D	A	C	B	A	tmp								
hist									tmp								

- 5) En utilisant la configuration suivante, donner l'état de **hist** et **tmp** pour retourner à Mozilla.

hist	A	F						tmp	A	B	C	A	D	E		
hist								tmp								

- 6) A partir de ces exemples donner une structure qui permette de conserver un historique des pages visitées et de naviguer dans les différentes pages internet.

Partie B : Implémentation de l'algorithme

voir le fichier [boutonweb.py](#)

Je n'ai pas réussi à implémenter correctement les boutons

Je laisse des commentaires si vous voulez le faire dans le fichier Python.