

Cours 26 novembre — Résumé de la séance

Pour une fois, nous avons pris le temps de respirer un peu durant cette séance. J'ai rapidement présenté les possibles poursuites d'études avec NSI (et Maths). A lire pour votre culture personnelle !

1) Poursuite d'études — 20 minutes

J'estime que cette interro contient exactement ce qu'il faut savoir sur les listes chaînées pour le BAC. Faites-la donc en autonomie et sans utiliser de ressources extérieures afin de voir ce que vous avez compris.

Le corrigé de l'interro est également disponible sur bouillotvincent.github.io . Relisez la. C'est un excellent moyen de vous tester et savoir si cette notion est comprise ou non.

2) Point sur les piles — 15 minutes

Retour sur les fonctions empiler, dépiler, est_vide, seules fonctions que l'on peut utiliser dans ce TD. Nous avons ensuite repris les algorithmes sur la question du parenthésage.

3) Parenthésage correct — 1h20

Vous devez impérativement essayer de faire la partie sur le parenthésage. L'algorithme est donné dans l'énoncé mais de manière très grossière.

Ne sautez donc pas l'étape du pseudo-code : c'est cette étape qui va vous faire raffiner le pseudo-code et vous permettre d'écrire votre programme Python.

Pensez aux cas particuliers/cas pratiques : est-ce que mon code prend en compte les cas particuliers ?

Par exemple :

ok : $(a+b)$

manque une parenthèse finale : $(a+(a+b)$

trop de parenthèses fermantes : $(a+(b+c))) + (c+d)$

De la même manière, on souhaite ajouter les $[]$.

De nouveaux cas apparaissent :

ok : $(a+[b+c])$

ok : $[a+(a+b)]$

trop de crochets ouvrants : $[[a+(a+b)]$

les crochets et les parenthèses sont imbriqués : $(a+[b+c])$

Un code corrigé vous est bien sûr proposé sur bouillotvincent.github.io .