

CAPES MATHS OPTION INFORMATIQUE

LISTE CHAINEES, PILES ET FILES

Classe Liste chaînée

Construire une classe implémentant les listes doublement chaînées non circulaires incluant :

- un constructeur
- deux attributs premier et dernier qui sont des cellules
- des méthodes `estVide`, `nbElements`, `iemeElement`, `modifierlemeElement`, `afficher`, `ajouterEnTete`, `ajouterEnQueue`, `supprimerTete`, `rechercherElement` et `insererElement`

Vous définirez pour cela une classe-structure `Cellule` composée de trois champs : `info`, `suivant` et `precedent`.

Classe Pile

Construire une classe implémentant les piles (sans utiliser le module `queue` de Python), incluant :

- un constructeur par copie
- un attribut liste (conteneur liste)
- la surcharge de la méthode d'affichage
- des méthodes `empiler`, `depiler`, `sommet`, `estVide`, `hauteur` et `pop`

Classe File

Construire une classe implémentant les files (sans utiliser le module `queue` de Python), incluant :

- un constructeur par copie
- un attribut liste (conteneur liste)
- la surcharge de la méthode d'affichage
- des méthodes `pop`, `push`, `estVide` et `longueur`

Inversion d'une File en utilisant une Pile

Le but de cet exercice est d'écrire en Python une procédure qui inverse une file d'entiers qui lui est passée en paramètre. On demande de ne pas utiliser de tableau ou de liste de travail pour effectuer l'inversion, mais d'utiliser plutôt une pile. Il existe en effet une méthode très simple pour inverser une file en utilisant une pile.

Validité du parenthésage d'une expression

Un problème fréquent pour les compilateurs et les traitements de textes est de déterminer si les parenthèses d'une chaîne de caractères sont équilibrées et proprement incluses les unes dans les autres. On désire donc écrire une fonction qui teste la validité du parenthésage d'une expression :

– on considère que les expressions suivantes sont valides : `"()"`, `"[[[bonjour+]essai]7plus-];"`

– alors que les suivantes ne le sont pas : `"(, ")"`, `"4(essai)"`.

Notre but est donc d'évaluer la validité d'une expression en ne considérant que ses parenthèses et ses crochets. On suppose que l'expression à tester est dans une chaîne de caractères, dont on peut ignorer tous les caractères autres que '(', '[', ']' et ')'. Écrire en Python la fonction `valide` qui renvoie vrai si l'expression passée en paramètre est valide, faux sinon.