

Faculté des Sciences de l'Ingénieur
Département d'Informatique
LMD G.L. Module SDA

Série d'exercices n°2

Exercice1

Soit la signature du TAD DATE suivant décrivant le type de donnée Date.

- 1- Distinguer entre les opérations internes et observateurs.
- 2- Les données suivantes sont elles correctes syntaxiquement selon la signature ci-dessous ?
12/09/90
12/09/1290
Année(12/mars/90)
Mois(12 avril 1990)
Ajouter_jours(12/09/90, 14)
Jour(Ajouter_jours(12/09/90, 56))
- 3- Ecrire un nombre suffisant d'axiomes pour ce TAD, sachant que les dates en question sont comprises entre 1900 et 2050.
- 4- Enrichir ce TAD avec l'opération **différence** qui calcule le nombre de jours écoulés entre deux dates.

TAD DATE

Sorte Date

Utilise NATUREL

Opérations :

//_ : Naturel, Naturel, Naturel → Date

*/*créer une date à partir de jour, mois, année*/*

année : Date → Naturel */*année de cette date*/*

mois : Date → Naturel */*no. du mois d'une date donnée*/*

jour : Date → Naturel */*no. du jour */*

ajouter_jours: Date, Naturel → Date

*/*la date n jours après */*

Exercice2

Ecrire le TAD Disque décrivant la structure de donnée disque formée à partir des coordonnées du centre et de son rayon. Prévoir l'opération **Surface** qui calcule la surface d'un disque

Exercice3

Soit la signature du TAD expression suivant :

TAD Expression
sorte expression
utilise VARIABLE, NATUREL
opérations
 var : variable → expression
 const : naturel → expression
 + : expression, expression → expression
 * : expression, expression → expression
 ° : variable, naturel → expression
 - : variable, naturel → expression

et nous rappelons les dérivées suivantes

$$\frac{d}{dx} x \equiv 1 \qquad \frac{d}{dx} x^n \equiv 0 \quad \forall n \in \mathbb{N}$$

$$\frac{d}{dx} (f+g) \equiv \left(\frac{d}{dx} f \right) + \left(\frac{d}{dx} g \right)$$

$$\frac{d}{dx} (f * g) \equiv \left(\frac{d}{dx} f \right) * g + \left(\frac{d}{dx} g \right) * f$$

$$\frac{d}{dx} x^n \equiv n x^{n-1} \quad \forall n \in \mathbb{N}$$

Enrichir le TAD expression avec l'opération dérivée

Exercice4

Enrichir le TAD liste par les opérations suivantes:

Queue, adjq (adjonction en queue), supq (suppression en queue), maximum, conc (concaténer), supp-pos (suppression de l'élément à une position donnée), est-vide, appartient, égale, triée.

Exercice5

- 1- Enrichir le TAD LISTE par l'opération : insérer-liste-triée qui insère un élément dans une liste triée par ordre croissant.

- 2- Ecrire la fonction C associée pour une représentation chaînée de la liste. Penser à la même question avec la représentation contiguë de la liste.

Exercice6

Soit le TAD liste récursive

- 1- Enrichir ce TAD LISTE avec l'opération inverse qui permet d'inverser une liste.
- 2- Ecrire une procédure récursive (algorithme) qui réalise cette opération.
- 3- Ecrire une procédure itérative (algorithme) qui réalise cette opération.
- 4- En utilisant la représentation chaînée, écrire une fonction C qui réalise cette opération.

Exercice7

Une liste l_1 est sous_liste d'une autre liste l_2 , si on peut obtenir l_1 à partir de l_2 , en effaçant des termes (on peut en effacer 0 ou tous les éléments). Par exemple, (5,3,9,4,15) est sous_liste de (89,5,2,5,3,9,6,10,4,15).

- 1- Enrichir le TAD LISTE par l'opération sous_liste qui teste si une liste l_1 est une sous liste d'une autre liste l_2 .
- 2- Ecrire la fonction récursive est_sous_liste (algorithme) qui prend deux listes en arguments et rend vrai ssi la première est sous_liste de la seconde.
- 3- En utilisant la représentation chaînée, écrire la fonction est_sous_liste en C qui prend deux listes en arguments et rend vrai ssi la première est sous_liste de la seconde.

Exercice8

Soient deux listes L1 et L2. Enrichir le TAD LISTE par l'opération **différence** qui détermine les éléments de la liste L1 qui n'appartiennent pas à la liste L2.

Exercice9

Montrer que les opérations du TAD PILE (cas particuliers des listes) : empiler, dépiler et sommet peuvent s'écrire en termes des opérations du TAD LISTE.

- 1- Montrer que les opérations du TAD FILE (cas particuliers aussi des listes) : ajouter, retirer et premier peuvent s'écrire en termes des opérations du TAD LISTE.

Exercice10

- 1- Enrichir le TAD PILE et FILE avec l'opération longueur qui calcule la longueur d'une pile ou d'une FILE.
- 2- Ecrire des fonctions C récursives qui retournent le nombre d'éléments d'une pile ou d'une file.
- 3- Ecrire des fonctions C itératives réalisant le même travail.

Exercice11

- 1- Enrichir le TAD PILE avec l'opération qui retourne l'élément maximum d'une pile
- 2- Ecrire une fonction C récursive qui retourne le plus grand élément de la pile.
- 3- Ecrire une fonction C itérative réalisant le même travail.
- 4- Refaire le même travail pour le cas d'une file