
Examen Final de Structure de données

Durée de l'examen : 1H30	Nombre de pages : 02
--------------------------	----------------------

Exercice N°01 : 7points = 1 + 1 + 2 + 3

Soient les deux fonctions suivantes :

Algorithme 1 Regardez_ces_fonctions

{ La première fonction }

```
fonction mystere1(n :entier) :entier
var resultat1 :entier ;
si n=0 alors
    resultat1 ← 0;
sinon
    resultat1 ← mystere1(n - 1) + (n * n);
fin
renvoyer(resultat1);
fin ;
```

{ La deuxième fonction }

```
fonction mystere2(b :entier ; n :entier) :entier ;
var resultat2 :entier ;
si n=0 alors
    resultat2 ← 1;
sinon
    resultat2 ← 1 + b * mystere2(b, n - 1);
fin
renvoyer(resultat2);
fin ;
```

1. Dérouler la fonction "mystere1" pour $n = 2$, pour $n = 3$.
2. Dérouler la fonction "mystere2" pour $(b = 2 \text{ et } n = 2)$, pour $(b = 2 \text{ et } n = 3)$.
3. En déduire ce que font ces fonctions.
4. Proposer deux fonctions itératives (sans la récursivité), qui font la même chose que les deux fonctions précédentes.

Exercice N°02 : 06 points = 2 + 2 + 2

Soient P une Pile d'entier, et F une File d'entier.

1. Écrire une fonction "*copier(P : Pile) : File*" qui copie les éléments de la pile P dans la file F.
2. Écrire une fonction "*copier_inverse(P : Pile) : File*" qui copie les éléments de la pile P dans la file F, mais cette fois dans l'ordre inverse.
3. Écrire une fonction "*test(P : Pile) : booléen*" qui renvoie "vrai" si la pile P est triée selon un ordre croissant et "faux" sinon.

☞ (Attention à ne pas perdre les éléments de P!).

Exercice N°03 : 08 points = 1 + 2 + 2 + 2

Algorithme exo2

type Liste=enregistrement

elt :entier;

suivant : \uparrow Liste;

fin_enregistrement

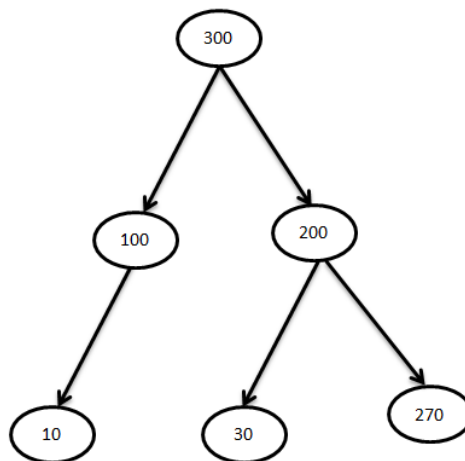
var T : \uparrow Liste;

Soit une liste linéaire chaînée d'entiers, supposée non vide, dont le pointeur vers la tête est T.

1. Écrire une fonction "*somme(T) : entier*" qui renvoie la somme des éléments de la liste.
2. Écrire une fonction "*test(T) : booléen*" qui renvoie "vrai" si la liste est triée selon un ordre croissant et "faux" sinon.
3. Écrire une procédure "*permute(T)*" qui permute le premier et le dernier élément de la liste.
4. Écrire une procédure "*tri(T)*" qui trie la liste selon un ordre décroissant.

Questions Bonus 02points = 0,5 + 0,5 + 1

Soit l'arbre suivant,



1. Cet arbre est-il binaire ?
2. Quelle est son hauteur ?
3. Donner deux fils et deux pères.

Mr. BEZOU

Bon courage !