

Examen Final LO21 A06

Durée: 2 heures

Documents non autorisés

Exercice 1

On souhaite développer un programme qui calcule la K ième dérivée d'un polynôme $P(X)$ de degré n . Pour cela, on considère que le polynôme $P(X)$ est la somme des monômes $M_i(X)$ tels que $M_i(X) = a_i * X^i$.

$$P(X) = a_0 + a_1 * X^1 + \dots + a_i * X^i + \dots + a_n * X^n = M_0(X) + M_1(X) + \dots + M_i(X) + \dots + M_n(X)$$

On définit la K ième dérivée de $P(X)$ que l'on note $D_k(P(X))$, comme étant le polynôme somme des K ièmes dérivées des monômes $M_i(X)$, que l'on note $D_k(M_i(X))$, c'est à dire :

$$D_k(P(X)) = D_k(M_0(X)) + D_k(M_1(X)) + \dots + D_k(M_i(X)) + \dots + D_k(M_n(X)).$$

On rappelle que la K ième dérivée d'un monôme $M_i(X)$, de degré i , est le monôme $D_k(M_i(X))$ défini de la manière suivante :

$$D_k(M_i(X)) = 0 \text{ si } k > i$$

$$D_k(M_i(X)) = i * (i-1) * \dots * (i-k+1) * a_i * X^{i-k} \text{ sinon.}$$

$$\text{Exemple : } D_1(M_i(X)) = 0 \text{ si } i > 1, D_1(M_i(X)) = i * a_i * X^{i-1} \text{ sinon.}$$

- 1- En utilisant la définition de $D_k(M_i(X))$, établir la relation de récurrence qui calcule la K ième dérivée d'un monôme en fonction de sa $(K-1)$ ième dérivée.
- 2- Un polynôme est considéré comme une liste de monômes.
 - Ecrire l'algorithme itératif du sous-programme qui calcule le polynôme contenant la k ième dérivée d'un polynôme.
 - Ecrire l'algorithme récursif du sous-programme qui calcule le polynôme contenant la k ième dérivée d'un polynôme.
- 3- On représente les polynômes par des listes chaînées de monômes. Donner en Langage C, la déclaration complète du type polynôme.

Exercice 2

1- Une tâche est caractérisée par un code (un entier) et un temps de réalisation (un réel). On souhaite construire un arbre binaire de tâches à réaliser par une machine de production (la valeur d'un nœud est une tâche) tel qu'en chaque nœud N de cet arbre :

- le temps de réalisation de la tâche du fils gauche de N est inférieur ou égal au temps de la tâche de N .
- le temps de réalisation de la tâche du fils droit de N est supérieur au temps de réalisation de la tâche de N

Ecrire l'algorithme du sous programme qui insère une tâche dans un arbre binaire de tâches.

2- Etant donné un temps de réalisation, écrire l'algorithme du sous programme qui calcule le nombre de tâches ayant un temps de réalisation inférieur ou égale à ce temps.

3- Donner en C : la déclaration complète du type arbre binaire de tâches, et la traduction du sous-programme de la question 1.