

Algorithmique II

Examen final

Durée : 1h 30mn

Exercice 1 : (Sur 6 points)

Un marchand de produits représente chaque Produit comme une structure contenant les champs suivants :

- CodeProduit, une chaîne de caractères de taille maximale 10 caractères, indiquant le code du Produit.
- NomProduit, une chaîne de caractères de taille maximale 20 caractères, indiquant le nom du Produit.
- PrixUnitaire, un réel indiquant le prix unitaire du produit.
- Quantité, un entier indiquant la quantité à vendre du produit.
- Stock, un entier indiquant la quantité du produit se trouvant dans le stock.

Pour établir la facture des produits vendus pour un client le marchand range la liste des produits vendus à ce client dans un tableau de Produits nommé Tab.

1. a) Donner la déclaration du type Produit en tant que structure regroupant les champs ci-dessus.
b) Donner la déclaration de Tab en tant que variable de type tableau de Produit, de taille N (N est une constante).
2. Ecrire une procédure AfficherFacture (Tab ; NomClient ; M : Entier) recevant le tableau Tab, une chaîne de caractères NomClient de taille maximale 20 caractères donnant le nom du client ainsi que le nombre M de produits vendus et la procédure affichera les produits du tableau, après avoir vérifié que pour chaque produit la quantité vendue est disponible en stock. L'affichage sera comme suit :

<u>Nom</u> : <u>NomClient</u>				
<u>Code du Produit</u>	<u>Nom du Produit</u>	<u>Prix Unitaire</u>	<u>Quantité</u>	<u>Prix Total</u>
.....
<u>Total</u> :				

"Prix Total" est le prix de la quantité vendue du produit, si elle est disponible dans le stock et
"Total" est la somme des prix totaux

Exercice 2 : (Sur 7 points)

Soit $Ch[1..n]$ et $P[1..m]$ deux chaînes de caractères, avec $1 \leq m \leq n$ On considère la fonction Match (P ; Ch ; m ; n) définie par :

Fonction Match (P : Caractere[1..m]; Ch : Caractere[1..n]) : Entier

Var L, r : Entier

B : Boolleen

Debut

$L \leftarrow 0$

$B \leftarrow \text{Faux}$

```

    Tant que ( $L \leq (n-m)$  Et  $B = \text{Faux}$ ) Faire
         $L \leftarrow L + 1$ 
         $r \leftarrow 1$ 
         $B \leftarrow \text{Vrai}$ 
        Tant que ( $r \leq m$  Et  $B = \text{Vrai}$ ) Faire
            Si ( $P[r] \neq \text{Ch}[L+r-1]$ ) Alors
                 $B \leftarrow \text{Faux}$ 
            Fin Si
             $r \leftarrow r + 1$ 
        Fin Tant que
        Si ( $B = \text{Vrai}$ ) Alors
            Retourner  $L$ 
        Fin Si
    Fin Tant que
    Retourner -1
Fin

```

1. Expliquer brièvement le but de la fonction *Match* ?
2. Calculez la complexité temporelle $C(m, n)$ dans les pires des cas de la fonction *Match*, ceci en fonction de m et n .
3. Modifiez la fonction *Match* de manière à ce qu'elle fournisse le nombre d'occurrences de P dans Ch (Les occurrences peuvent se chevaucher).

Exercice 3 : (Sur 7 points)

On considère la suite récurrente donnée par :

$$a_0=1 ; a_1=2 \text{ et } a_n = a_{n-1} * a_{n-2} \text{ pour } n > 1$$

1. Ecrire en pseudo-code une fonction récursive *SuiteRecursive* () qui retourne le $n^{\text{ème}}$ élément de cette suite.
2. Ecrire en pseudo-code une fonction itérative *SuiteIterative* () qui retourne le $n^{\text{ème}}$ élément de cette suite.
3. Laquelle des deux fonctions est plus efficace. Justifier votre réponse en calculant les complexités temporelles des deux fonctions.

Bon courage