

Algorithmique II

Examen de rattrapage

Durée : 1h 30 mn

Exercice 1 : (Sur 8 points)

On souhaite écrire un algorithme qui permet de gérer un ensemble d'étudiants. Chaque type Etudiant est une structure composée des champs suivants :

- Nom, une chaîne de caractères
- DateNaissance, un tableau de trois entiers (jour, mois et année)
- MoyenneGénérale, un réel

1. Donner le type Etudiant et déclarer le tableau Tab, comme variable globale, constituée de N Etudiant, où N est une constante donnée.
2. Écrire les fonctions et procédures qui permettent de réaliser les tâches suivantes :
 - a) La fonction *Créer(nom : chaîne ; j, m, an : Entier ; moy : Réel)* qui permet de créer une variable de type Etudiant, saisir le nom de cet étudiant (nom), sa date de naissance (j, m, an), sa moyenne (moy) et retourner la valeur de la variable ainsi créée.
 - b) La procédure *SaisirTableau()* qui permet de remplir le tableau Tab déclaré ci-dessus, en lisant les données de chaque étudiant, et en utilisant la fonction *Créer*, définie ci-dessus.
 - c) La procédure *CalculeMoyenne()* qui permet de calculer et afficher la moyenne de la classe constituée des étudiants du tableau Tab.

Exercice 2 : (Sur 5 points)

Pour déterminer et afficher les N premiers nombres premiers c'est à dire la suite : 2, 3, 5, 7, 11, ... (où N est une constante donnée), on utilise l'algorithme *nombresPremiers ()* suivant :

On utilise un tableau d'entiers T[1 .. N], dans lequel on range les nombres premiers découverts au fur et à mesure.

On note par m le nombre d'entiers premiers connus et rangés dans T à une étape donnée du déroulement de l'algorithme.

On note par k le plus grand entier dont on a testé la primalité.

1. On considère les affectations suivantes : $T[1] \leftarrow 2$, $m \leftarrow 2$ et $k \leftarrow 2$.
2. On répète ce qui suit tant que $m \leq N$.
 - On incrémente k ($k \leftarrow k + 1$).
 - Si k n'est divisible par aucun des éléments T[1], ..., T[m-1], alors k est premier, et dans ce cas on affecte k à T[m] ($T[m] \leftarrow k$) et on incrémente m ($m \leftarrow m + 1$) ; Sinon on retourne à l'étape 2.
3. On affiche tous les éléments du tableau T.

Ecrire en pseudo-code l'algorithme *nombresPremiers ()*, décrit ci-dessus.

Exercice 3 : (Sur 7 points)

On considère la fonction *vérifier* (*Ch* : Chaîne, *deb* : Entier, *fin* : Entier) donnée par :

Fonction *vérifier* (*Ch* : Chaîne, *deb* : Entier, *fin* : Entier) : *booléen*

Var *i, j* : Entier

Début

 Si (*deb* >= *fin*) Alors

 Retourner (Vrai)

 Sinon

 Si (*Ch*[*deb*] <> *Ch*[*fin*]) Alors

 Retourner (Faux)

 Sinon

 Retourner (*vérifier*(*Ch*, *deb*+1, *fin*-1))

 Fin Si

 Fin Si

Fin

1. Appeler la fonction *vérifier*(*Ch*, *deb*, *fin*) dans le cas suivant :

Ch = "bonjob", *deb*=0 et *fin*=5

 Et donner la valeur retournée par cet appel.

2. Quel est le but de la fonction *vérifier*(*Ch*, 0, *n*-1), où *n* est la longueur de la chaîne *Ch* ?

3. Déterminer la complexité temporelle de la fonction *vérifier*(*Ch*, 0, *n*-1) en fonction de *n*.