

TD Algorithmique et Structure de Données 2 -S2-

Série 03

Exercice 03.

- **Énoncé:** Soit une section de plusieurs élèves dont le nombre ne dépasse pas 100 (*c.-à-d. le nombre maximum d'étudiants = 100*). On sauvegarde les informations concernant chaque élève dans un tableau (*tab_élèves: alors. le tableau « tab_élèves [] » est de taille max 100*).
- Ecrire un programme qui saisit (*scanf()*) plusieurs élèves (nom, prénom, âge: *on définit une structure « étudiant » possédant trois objets*) et les range dans un tableau (nom = "fin" détermine la fin de la sauvegarde) (*on définit un tableau « tab_élèves [] » de type structure « étudiant »*). Faire la sauvegarde du contenu du tableau dans un fichier texte (*le mode d'ouverture du fichier = 'w', c.-à-d. on écrit seulement dans le fichier...*) dont le nom sera demandé à l'utilisateur (*scanf()*). Chaque élève sera sauvegardé sur une ligne du fichier (*c.-à-d. on utilise \n*). Chaque champ d'une même ligne est séparé de l'autre par un espace ().

Solution: EX03.

- ❖ #include <stdio.h>
- ❖ #include <string.h> // permet d'effectuer plusieurs opérations sur des chaînes de caractères (ex. **strcmp()**)
- ❖ #define TAILLE_NOM 20 // on associe la valeur **20** à la constante **TAILLE_NOM**.
- ❖ #define TAILLE_PRENOM 2*TAILLE_NOM // c.-à-d. **TAILLE_PRENOM** reçoit la valeur constante **2*20**.
- ❖ #define MAX_ETUDIANTS 100 // la constante **MAX_ETUDIANTS** contient la valeur 100.
- ❖ main () // la fonction principale est l'endroit où un programme commence son exécution.
- ❖ { //début du programme
- ❖ char nomfichier[20] ; //définir une chaîne de caractères « **nomfichier** » de taille 20 « nombre de caractères » contiendra le nom du fichier saisi par l'utilisateur.
- ❖ FILE *fichier ; // créer un pointeur « fichier » de type **FILE** contiendra l'adresse du début du fichier.
- ❖ int i, nbelevés, finir_saisie ; // déclaration de variables entières.
- ❖ struct etudiant // définir une structure « étudiant » à l'aide du mot clé **struct** qui se compose de 3 objets:
- ❖ {
char nom[TAILLE_NOM] ; // **nom [20]** est de type chaîne de caractères « **char** »
char prenom[TAILLE_PRENOM] ; // **prenom[2*20]** est de type chaîne de caractères « **char** »
int age ; // age : entier
- ❖ } ;

Solution: EX03.

- ❖ `struct etudiant tab_eleves[MAX_ETUDIANTS]; // définir tab_eleves[100] de type structure « étudiant », pour accéder au champ nom de la structure « étudiant » du premier élève se réalise comme suit: tab_eleves [0] . nom;`
- ❖ `i=0 ; finir_saisie = 0 ; //initialisation`
- ❖ `while (!finir_saisie) // while (!0): tant que la condition est vraie (la saisie n'est pas terminée ,finir_saisie #1) faire :`
- ❖ `{ // pour lire une chaîne de caractères;`
- ❖ `scanf("%s",tab_eleves[i].nom); // par ex. tab_eleves[0].nom; lire le nom du premier élève.`
- ❖ `finir_saisie=((strcmp(tab_eleves[i].nom,"fin"))==0); // La fonction strcmp () compare deux chaînes (tab_eleves[i].nom et 'fin') et renvoie 0 si les deux chaînes sont identiques; cette instruction représente la fin de remplissage du tableau, c.-à-d.. si la lecture du nom est égale à 'fin' , alors la fonction retourne 0 et la variable entière finir_saisie reçoit 1. (sinon elle reçoit 0)`
- ❖ `if (!finir_saisie) // on continue à lire le prénom et l'âge si la condition est vérifiée (vraie: if(!0))`
- ❖ `{`
- ❖ `scanf("%s",tab_eleves[i].prenom) ; //par ex. tab_eleves[0].prenom; lire le prénom du premier élève`
- ❖ `scanf("%s",&tab_eleves[i].age) ; //par ex. tab_eleves[0].age; lire l'âge du premier élève`
- ❖ `i++; //incréméntation de la boucle while() pour lire les données du prochain étudiant, également la variable « i » joue le rôle d'un compteur pour calculer le nombre d'élèves sauvegarder dans le tableau « tab_eleves [i]»`
- ❖ `}`
- ❖ `}`

Solution: EX03.

- ❖ `nbelevés=i+1; // la variable nbelevés reçoit la valeur du compteur i +1, car le tableau commence par l'indice i=0;`
- ❖ `printf("Donner le nom du fichier "); // affichage pour l'utilisateur.`
- ❖ `scanf("%s", nomfichier); // lire le nom du fichier; on utilise le symbole %s car la variable « nomfichier » est du type chaîne de caractères défini ci-dessus.`
// Ouverture du fichier en écriture grâce à « w »
- ❖ `fichier = fopen(nomfichier,"w"); // la variable « fichier » est de type FILE*; pour ouvrir un fichier on utilise la fonction fopen() qui comprend deux paramètres: le nom du fichier « nomfichier » à ouvrir et le mode d'ouverture du fichier « w »;`
« w » : crée un fichier vide pour l'écriture. Si un fichier du même nom existe déjà, son contenu est effacé et le fichier est considéré comme un nouveau fichier vide.
- ❖ `for (i=0 ; i<nbelevés;i++) // la boucle for() pour parcourir le tableau contenant les informations concernant les étudiants (après chaque itération, la boucle est incrémentée de +1 jusqu'à atteindre la dernière case du tableau représentant le dernier élève inscrit); tant que i est strictement inférieur au nombre d'élèves inscrits dans le tableau « tab_eleves [] » faire:`
- ❖ `{ // la fonction fprintf() contient presque la même syntaxe que la fonction printf ("%s",tab_eleves[i].nom), sauf qu'elle est utilisée pour écrire dans un fichier « FILE* fichier » défini comme le premier paramètre de la fonction.`
- ❖ `fprintf(fichier,"%s",tab_eleves[i].nom);`
`fprintf(fichier,"%s",tab_eleves[i].prenom);`
`fprintf(fichier,"%d\n",tab_eleves[i].age);`
`}`

ici il y a un espace

*// on récupère les informations des élèves dans le tableau « tab_eleves [] » et on les écrit ligne par ligne dans le fichier « fichier » en utilisant **\n**.*
- ❖ `fclose(fichier); // après avoir utilisé le fichier « fichier », il faudra le fermer pour libérer de la mémoire, pour cela on emploie la fonction fclose (FILE*)`
- ❖ `} // fin du programme`