

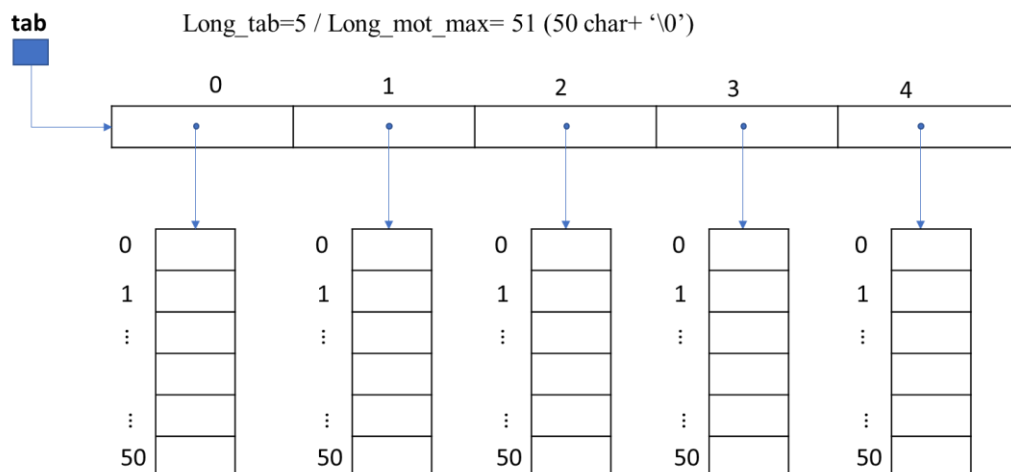
Travaux Pratiques N°4 (TP 4)

(Allocation dynamique et listes chaînées)

Exercice 1

Écrire un programme C qui comprend les fonctions suivantes :

- 1) Une fonction « Saisie » qui permet de lire 5 mots d'une longueur maximale de 50 caractères. Ces mots doivent être stockés dans un tableau de chaînes de caractères, avec une allocation dynamique de mémoire.
- 2) Une fonction « insertionMot » qui permet d'insérer un nouveau mot dans le tableau des chaînes de caractères.
- 3) Une fonction « inverserMot » qui permet d'inverser l'ordre des caractères d'une chaîne en utilisant deux pointeurs P1 et P2.
- 4) Une fonction « inverserTab » qui permet d'inverser à la fois les mots individuels et l'ordre global des mots dans un tableau de caractères en utilisant deux pointeurs P1 et P2.
- 5) Une fonction « Afficher » qui permet d'afficher le contenu du tableau.
- 6) Une fonction « Libérer » qui permet de libérer l'espace mémoire alloué pour le tableau.
- 7) Une fonction principale « main » permettant d'exploiter ces fonctions.



Exercice 2

Écrire un programme C qui comprend les fonctions suivantes :

- 1) Une fonction « Saisie » qui permet de lire **n** entiers entré par l'utilisateur. Ces mots doivent être stockés dans un tableau **T**, avec une allocation dynamique de mémoire.
- 2) Une fonction « Afficher » qui permet d'afficher le contenu du tableau **T**.
- 3) Une fonction « triSelection » qui permet de trier le tableau **T** par ordre croissant.
- 4) Une fonction principale « main » permettant d'exploiter ces fonctions.

Exercice 3

Ecrire un programme C qui permet de :

- 1) Une fonction « **Saisie** » qui permet de lire 5 mots d'une longueur maximale de 50 caractères. Ces mots doivent être stockés dans un tableau de pointeurs vers char nommé **CH**, avec une allocation dynamique de mémoire.
- 2) Une fonction « Copier » permettant de concaténer les 5 mots en une seule chaîne de caractères (une phrase), en les copiant progressivement. L'adresse de cette phrase doit être stockée dans un pointeur nommé **PHRASE**. Avant la copie, il est nécessaire de réserver dynamiquement l'espace mémoire requis pour contenir la phrase complète (en prenant en compte les espaces éventuels et le caractère de fin de chaîne).
- 3) Une fonction « liberer » qui permet de libérer les différentes zones mémoires allouée dynamiquement.
- 4) Une fonction principale « main » permettant d'exploiter ces fonctions.

Exercice 4

On veut gérer une liste chaînée simple d'étudiants. Chaque **Cellule** (noeud) représente un étudiant qui est définie par :

- **NOM** : le nom de l'étudiant (chaîne de caractères).
- **PRENOM** : le prénom de l'étudiant (chaîne de caractères).
- **MOYENNE** : la moyenne générale de l'étudiant (réel).

- 1) Définir la structure **Cellule**.
- 2) Définir une fonction « saisieEtudiant » qui permet de saisir les informations d'un étudiant. Cela revient à remplir les différents champs de la structure 'Cellule'.

- 3) Définir une fonction « affichageEtudiant » qui permet d'afficher les informations d'un étudiant. Cela revient à afficher les différents champs de la structure 'Cellule'.
- 4) Définir une fonction « insererTete » qui permet d'insérer un étudiant donné en tête de la liste.
- 5) Définir une fonction « affichageListe » qui permet d'afficher les informations de tous les étudiants dans la liste.
- 6) Définir une fonction « rechercheEtudiant » qui permet de rechercher un étudiant par son nom et prénom. Si l'étudiant est trouvé la fonction retourne l'adresse de la cellule contenant l'étudiant recherché, sinon la fonction retourne NULL.
- 7) Définir une fonction « TriListe » qui permet de trier la liste des étudiants par ordre croissant de moyenne.
- 8) Définir une fonction « supprimerEtudiant » qui permet de supprimer un étudiant par son nom et prénom. La fonction retourne la liste mise à jour après la suppression.
- 9) Définir une fonction « libererListe » qui permet de libérer l'espace mémoire allouée pour la liste des étudiants.
- 10) Une fonction principale « main » permettant d'exploiter ces fonctions.