Univérsité Hassan II Casablanca

Ecole normale supérieure (ENS) Casablanca

Structures de données en C

Année universitaire 2020/2021

Parcours 2éme année LEESM **Prof. M. Lahby**

TP 3

Gestion dynamique de la mémoire

Exercice 1

Ecrire un programme qui lit 10 phrases d'une longueur maximale de 200 caractères au clavier et qui les mémorise dans un tableau de pointeurs sur char en réservant dynamiquement l'emplacement en mémoire pour les chaînes. Ensuite, l'ordre des phrases est inversé en modifiant les pointeurs et le tableau résultant est affiché.

Exercice 2

Ecrire un programme qui lit 10 mots au clavier (longueur maximale : 50 caractères) et attribue leurs adresses à un tableau de pointeurs MOT. Effacer les 10 mots un à un, en suivant l'ordre lexicographique et en libérant leur espace en mémoire. Afficher à chaque fois les mots restants en attendant la confirmation de l'utilisateur (par 'Enter').

Exercice 3

Ecrire un programme qui lit 10 mots au clavier (longueur maximale : 50 caractères) et attribue leurs adresses à un tableau de pointeurs MOT. Copier les mots selon l'ordre lexicographique en une seule 'phrase' dont l'adresse est affectée à un pointeur PHRASE. Réserver l'espace nécessaire à la PHRASE avant de copier les mots. Libérer la mémoire occupée par chaque mot après l'avoir copié. Utiliser les fonctions de <string>.

Problème 1 : *"la gestion d'annuaire téléphonique" (version statique)*

On souhaite créer un programme en langage C permettant la gestion d'annuaire téléphonique. Cet Annuaire contient des renseignements téléphoniques sur les individus. On utilisera les deux types de données définis ci-dessous pour représenter un individu et un annuaire :

```
typedef struct {
char nom[15];
int annee_naissance;
char tel[20];
} individu;
```

Pour résoudre ce problème nous utilisons le mécanisme de l'allocation statique de la mémoire. Pour cela, on propose d'utiliser deux champs pour implémenter la structure carnet qui sont : un tableau statique **P** permettant de stocker les personnes (100 maximum) et un compteur n indiquant le nombre de personnes dans ce tableau statique.

```
typedef struct Carnet {
individu P[100];
int n;
}annuaire;
```

Q1. Ecrire une fonction d'entête individu saisir_individu(void) saisissant au clavier les données constituant un objet de type individu. Proposez un autre prototype pour cette fonction.

Q2. Ecrire une fonction d'entête int identiques(individu m1,individu m2) retournant 1 si les champs nom (considéré comme une chaîne de caractères) et annee_naissance de m1 et m2 ont les mêmes valeurs, et 0 sinon.

Q3. Ecrire une fonction d'entête void ajouter_personne(individu I, annuaire A) qui ajoute l'individu I dans l'annuaire A (on insère l'élément à la fin du tableau).

Q4. Ecrire une fonction d'entête void ajouter_personne(individu I, annuaire *A) qui permet d'ajouter l'individu I dans l'annuaire A (on insère l'élément à la fin)

Q5. Ecrire une fonction d'entête void afficher_individu(individu I) qui affiche les informations contenues dans l'objet I passée en argument.

Q6. Ecrire une fonction d'entête void afficher_annuaire(annuaire A) affichant à l'écran une représentation lisible de toutes les personnes de l'annuaire.

Q7. Ecrire une fonction d'entête int position(individu m, annuaire A) retournant l'indice où l'on trouve m dans A (-1 si m n'est pas dans l'annuaire).

Q8. Ecrire une fonction d'entête char *chercher_nom(char *T, annuaire A) retournant le nom de l'individu dont le numéro de téléphone est T dans l'annuaire A.

Q9. Ecrire une fonction d'entête char *chercher_tel(individu m, annuaire A) retournant le numero de téléphone de l'individu m dans l'annuaire A.

Q10. Ecrire une fonction d'entête void supprimer_personne(individu m, int p) qui supprime une personne du l'annuaire dont la position p est passé en argument.

Q11. Ecrire une fonction d'entête Trier_personne(annuaire *A) qui trie les personnes de l'annuaire.

Q12. Dans la fonction main(), créer un menu sous la forme suivante :

- **[1]** Ajouter une personne au Carnet
- **[2]** Supprimer une personne au Carnet
- **[3]** Affichage du carnet
- **[4]** Recherche d'une personne dans au Carnet
- **[5]** Trier les personne dans un carnet par ordre alphabétique
- **[6]** Quitter

Problème 2 : *"la gestion d'annuaire téléphonique" (version dynamique)*

En raison que l'allocation statique représente une perte de l'espace mémoire, on propose d'utiliser deux champs pour implémenter la structure carnet qui sont : un tableau dynamique permettant de stocker les personnes et un compteur n indiquant le nombre de personnes dans ce tableau dynamique.

```
typedef struct {
char nom[15] ;
int annee_naissance ;
char tel[20] ;
} individu ;

typedef struct Carnet {
individu *P ;
int n ;
}annuaire ;
```

Question : refaire toutes les questions du problème 1.