

## TRAVAUX DIRIGES N°3

### Mémoire dynamique

#### Thèmes

- Structuration et généralisation
- Définition et documentation de fonctions

Les définitions et les codes seront exprimés en C++, le différentiel en C sera donné pour les questions 1.3 et 1.5.

#### Exercice 1. Mémoire dynamique

Soit la structure suivante :

```
struct Personne {  
    char* nom;  
    int age;  
};
```

Cette structure va nous servir à illustrer l'allocation dynamique de mémoire. Le programme qui se trouve en annexe est incomplet. A chaque question qui suit correspond un extrait de code manquant. Dans le programme, un commentaire vous indique les numéros des questions correspondantes.

**Question 1.1** - Le programme va permettre la manipulation d'un tableau de pointeurs vers des données de type `Personne`. La taille de ce tableau est statique (indiquée par une constante nommée `MAX_PERSONNES`). Déclarez un tel tableau (nommé `personnes`). *Voir la fonction main.*

**Question 1.2** - Dans ce programme, nous indiquerons qu'une case d'un tableau ne désigne aucune personne en lui affectant la valeur nulle. Ecrire une fonction, nommée `initialiser`, qui initialise un tel tableau de manière à ce qu'aucune personne ne soit référencée. Cette fonction est invoquée dans la fonction `main`.

**Question 1.3** - Dans la fonction `ajouter`, après avoir demandé à l'utilisateur l'indice où ajouter la nouvelle personne et vérifié la validité de cet indice, il est nécessaire :

- d'allouer dynamiquement une structure de type `Personne`,
- d'initialiser la case du tableau,
- puis d'initialiser les champs de la structure en saisissant le nom et l'âge.

Complétez le code de la fonction `ajouter`.

*La fonction `ajouter` est invoquée dans la fonction `main`.*

**Question 1.4** - Complétez la fonction `afficher`.

*Cette fonction est invoquée dans la fonction `main`.*

**Question 1.5** - Complétez le code de la fonction `supprimer` et `desallouer` sachant que leurs rôles respectifs sont :

- de supprimer la personne désignée par l'indice choisi par l'utilisateur,
- de supprimer toutes les personnes référencées dans le tableau.

*Ces fonctions sont invoquées dans la fonction `main`.*

## Annexe de code

```
#include <iostream>
using namespace std;

struct Personne {
    char* nom;
    int age;
};

/* Question 1.2 - Programmer une fonction nommée initialiser */

/**
 * @brief Ajoute une Personne à un tableau de pointeurs de Personne
 *         l'indice de la personne à ajouter est saisi et contrôlé
 * @param[in,out] le tableau
 * @param[in] la taille du tableau
 */
void ajouter(Personne** personnes, unsigned int taille) {
    unsigned int i;
    cout << "numéro de la personnes à ajouter : ";
    cin >> i;
    if (i < 0 || i >= taille || personnes[i] != 0)
        cerr << "indice non valable ou déjà affecté" << endl;
    else {
        const int MAX_NOM = 50;
        char buffer[MAX_NOM];

        /* Question 1.3 - Allouer une nouvelle personne
           et initialiser ses champs */

    }
}

/**
 * @brief Affiche un tableau de pointeurs de Personne
 * @param[in] le tableau
 * @param[in] la taille du tableau
 */
void afficher(const Personne* const * personnes, unsigned int taille) {
    unsigned int i;
    for (i = 0; i < taille; ++i) {
        cout << '[' << i << " ] ";
        if (personnes[i])

            /* Question 1.4 - Afficher les données pointées par personnes[i] */

        else
            cout << "vide";
        cout << endl;
    }
}
```

```
/**
 * @brief Supprime une Personne d'un tableau de pointeurs de Personne
 *         l'indice de la personne à supprimer est saisi et contrôlé
 * @param[in,out] le tableau
 * @param[in] la taille du tableau
 */
void supprimer(Personne** personnes, unsigned int taille) {
    unsigned int i;
    cout << "numero de la personnes a supprimer : ";
    cin >> i;
    if (i < 0 || i >= taille || personnes[i] == 0)
        cerr << "indice non valable ou non affecte" << endl;
    else {

        /* Question 1.5 - Desallouer les données pointées par personnes[i] */

    }
}

/**
 * @brief Desalloue les postes d'un tableau de pointeurs de Personne
 * @param[in,out] le tableau
 * @param[in] la taille du tableau
 */
void desallouer(Personne** personnes, unsigned int taille) {
    unsigned int i;
    for (i = 0; i < taille; ++i) {

        /* Question 1.5 - Désallouer les données pointées par personnes[i] */

    }
}

int main() {
    const unsigned int MAX_PERSONNES = 5;

    /* Question 1.1 - Déclarer un tableau de pointeurs de Personne */

    initialiser(personnes, MAX_PERSONNES);
    int choix;
    do {
        cout << endl;
        cout << "0: quitter" << endl;
        cout << "1: afficher les personnes" << endl;
        cout << "2: ajouter une personne" << endl;
        cout << "3: supprimer une personne" << endl;
        cin >> choix;
        switch (choix) {
            case 1 : afficher(personnes, MAX_PERSONNES); break;
            case 2 : ajouter(personnes, MAX_PERSONNES); break;
            case 3 : supprimer(personnes, MAX_PERSONNES); break;
        }
    } while (choix != 0);
    desallouer(personnes, MAX_PERSONNES);
    return 0;
}
```