Travaux dirigés 6 : tableaux, chaînes, structures

1 Tableaux en C

Un tableau de variables est une structure de données permettant de déclarer une collection de variables de même type, dans une zone contigüe en mémoire. Ces variables sont référencées par l'indice de leur place dans le tableau. Donner un tableau en argument à une fonction pose des problèmes inutiles (allocation mémoire et manipulation de pointeurs), nous éviterons de le faire dans cette section, pour nous concentrer sur les manipulations de base en C puis sur l'écriture d'algorithmes dans la section suivante.

1.1 Affichage des éléments d'un tableau

Écrire un programme qui :
- déclare et initialise le tableau de variables entières : 5,2,4,3,0
- pour chaque case du tableau :
  - affiche la case.

Correction.

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int tab[5] = {5,2,4,3,0}; /* tableau a afficher */
    int i; /* var. de boucle */

    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* affichage de la case */
        printf("tab[%d] = %d\n",i,tab[i]);
    }
    /* i >= 5 */

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
```

1.2 Exercice type : initialisation des cases d'un tableau à zéro

Soit un tableau d'entiers, déclaré d'une taille quelconque. Écrire un programme qui :
- affiche le tableau non initialisé
- initialise chaque case à zéro
- affiche le tableau initialisé

Un exemple d'exécution pour un tableau de taille 3 est :

```
affichage du tableau non initialise :
tab[0] = 134513308
tab[1] = -1208832012
tab[2] = 134518316
affichage du tableau initialise :
tab[0] = 0
tab[1] = 0
tab[2] = 0
```

Correction. _____ *L'exemple plus haut correspond à l'exécution sur un PC 32 bits. Bien insister sur les valeurs arbitraires des variables non initialisées, qui dépendent des programmes qui ont été exécutés avant etc. On les représente par le symbole '?' dans la trace.*

```
algo:
- affichage du tableau non initialise
- pour chaque case du tableau :
  - affecte 0 a la case
- affichage du tableau initialise
```

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int tab[5]; /* tableau a initialiser */
    int i; /* var. de boucle */

    /* affichage du tableau non initialise */
    printf("affichage du tableau non initialise :\n");
    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* affichage de la case */
        printf("tab[%d] = %d\n",i,tab[i]);
    }
    /* i >= 5 */

    /* initialisation des cases a 0 */
    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* initialisation de la case a 0 */
        tab[i] = 0;
    }
```

```
    /* i >= 5 */

    /* affichage du tableau initialise */
    printf("affichage du tableau initialise :\n");
    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* affichage de la case */
        printf("tab[%d] = %d\n",i,tab[i]);
    }
    /* i >= 5 */

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
```

1.3 Exercice type : calcul de la somme des éléments d'un tableau

Soit un tableau d'entiers, initialisé à une taille et des valeurs quelconques. Écrire un programme qui calcule et affiche à l'écran la somme des éléments du tableau.

Correction.

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int tab[5] = {2,-4,8,12,-1};
    int somme = 0; /* element neutre pour l'addition */
    int i; /* var. de boucle */

    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* ajoute la case a la somme partielle */
        somme = somme + tab[i];
    }
    /* i >= 5 */

    /* somme vaut 0 + 2 - 4 + 8 + 12 - 1 */
    printf("somme = %d\n",somme);

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
```

1.4 Exercice type : calcul du minimum d'un tableau

Soit un tableau d'entiers, initialisé à une taille et des valeurs quelconques. Écrire un programme qui calcule et affiche à l'écran la valeur minimum des éléments du tableau.

Correction.

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */
#include <limits.h> /* INT_MAX */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int tab[5] = {2,-4,8,12,-1};
    int min = INT_MAX; /* +infini >= valeurs tableau */
    int i; /* var. de boucle */

    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        if(tab[i] < min) /* plus petit que min courant */
        {
            /* nouveau min courant */
            min = tab[i];
        }
    }
    /* i >= 5 */

    /* min contient le minimum de tous les elements du tableau */
    printf("le minimum des elements du tableau est : %d\n",min);

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
```

On peut aussi utiliser une fonction minimum.

```
/* declaration de fonctionnalites supplementaires */
#include <stdlib.h> /* EXIT_SUCCESS */
#include <stdio.h> /* printf */
#include <limits.h> /* INT_MAX */

/* declaration constantes et types utilisateurs */

/* declaration de fonctions utilisateurs */

/* fonction principale */
int main()
{
    /* declaration et initialisation variables */
    int tab[5] = {2,-4,8,12,-1};
```

```
    int min = INT_MAX; /* +infini >= valeurs tableau */
    int i; /* var. de boucle */

    for(i = 0;i < 5;i = i + 1) /* chaque case du tableau */
    {
        /* nouveau min courant */
        min = minimum(tab[i], min);
    }
    /* i >= 5 */

    /* min contient le minimum de tous les elements du tableau */
    printf("le minimum des elements du tableau est : %d\n",min);

    return EXIT_SUCCESS;
}

/* definitions des fonctions utilisateurs */
```

---

2 Trouver les bons algorithmes

Sans programmer vos solutions, donner des algorithmes pour les problèmes suivants. La valeur calculée (le produit, le nombre d'occurrences, l'indice, un booléen, etc.) sera simplement mise dans une variable résultat. Vous pouvez considérer que chaque tableau `t` a une taille `taille(t)`.

Calcul du produit des éléments d'un tableau. Soit un tableau d'entiers. Calculer le produit des éléments du tableau.

Calcul du maximum des éléments d'un tableau. Soit un tableau d'entiers. Calculer le maximum des éléments du tableau.

Calcul de la moyenne des éléments d'un tableau. Soit un tableau d'entiers. Calculer la moyenne des éléments du tableau.

Permutation circulaire d'un tableau. Soit un tableau, permuter ses éléments vers la droite : le premier devient le second, le second le troisième, ..., le dernier devient le premier.

Calcul du nombre d'occurrences d'un élément dans un tableau. Soit un tableau d'entiers et soit un entier n. Compter le nombre d'occurrences de n dans le tableau.

Unicité du premier élément d'un tableau. Soit un tableau d'entiers t. Tester si son premier élément est unique (il n'apparaît qu'une seule fois dans le tableau).

Unicité de tous les éléments d'un tableau. Soit un tableau d'entiers t. Tester si le tableau est sans répétitions (chacun de ses éléments est unique).

Tableau trié. Soit un tableau d'entiers. Tester si les éléments du tableau sont ordonnés par ordre croissant.

Recherche d'un entier dans un tableau. Soit un tableau d'entiers et soit un entier n. Trouver l'indice de la première occurrence de n dans le tableau (mettre cet indice à -1 si n n'apparaît pas dans le tableau).

Recherche d'un entier dans un tableau ordonné. Soit un tableau d'entiers, sans répétitions, classés par ordre croissant et soit un entier n. Trouver l'indice de n dans le tableau (mettre cet indice à -1 si n n'apparaît pas dans le tableau). Utiliser l'ordre pour trouver la réponse plus rapidement qu'à la question précédente.

Comparaison de tableaux. Soient deux tableaux de même taille. Vérifier s'ils sont égaux.

Un élément au hasard. Soit un tableau t donner un de ses éléments au hasard.

Mélanger un tableau. Soit un tableau t sans répétition, modifier au hasard l'ordre des éléments de t.

Tirage de p éléments parmi n. Soit deux entiers positifs p et n avec $p \leq n$. Donner au hasard une liste de p entiers, **sans répétition**, parmi $1, \ldots, n$.

2.1 Algorithmes sur du texte

Longueur d'une chaîne. Soit une chaîne de caractères s, déterminer le nombre de lettres qui la compose (on ne compte pas la sentinelle `\0`).

Comparaison de mots. Soient deux mots, stockés dans deux chaînes de caractères différentes. Comparer ces deux mots selon l'ordre lexicographique (l'ordre du dictionnaire). Le résultat prendra trois valeurs : 0 si les deux mots sont égaux, 1 si le premier mot est le premier dans l'ordre lexicographique, 2 si c'est le second.

Palindrome. Soit une chaîne de caractère s de n lettres. Tester si s est un palindrome. Vous pourrez distinguer deux cas : n pair et n impair.

Anagramme 1 (garemaman). Afficher la liste de toutes les combinaisons de lettres de la chaîne `ecran` (120 mots).

Anagramme 2. Soit une chaîne de caractère s de longueur n. Donner au hasard un anagramme de s.

Anagramme 3. Soit deux chaînes de caractère s et t. Tester si t est un anagramme de s.

Anagramme d'un palindrome. Soit une chaîne de caractères s. Tester si s est l'anagramme d'un palindrome (question posée en entretien d'embauche d'une grande entreprise d'informatique).

Non imorpta in che oridne apapaino le letetre in una paolra, l'uinca csoa imnorptate è che la pimra e l'ulimta letetra sinao nel ptoso gituso.