

Programmation C

Tableaux Statiques

ING1

CY Tech

Définition de constantes

Rappel

Constantes symboliques

- Remplacer une "expression" par un symbole dans le code
- Permet de clarifier le code
- `#define symbole expression`
 - ▶ `#define N 10`
 - ▶ `#define ERREUR_SAISIE -1`
 - ▶ `#define M 20;`
- Permet de déclarer une macro

Utilisation avancée

Macro

- Possibilité de déclarer un traitement spécifique
- Attention aux effets de bord ...
- `#define formule(a,b,...) expression`
 - ▶ `#define MAX(a,b) a>b?a:b`
 - ▶ `#define ERREUR(message) fprintf(stderr, "%s\n", message)`
- Utilisation des majuscules
- Éviter de mettre un ; à la fin d'un define

Vecteurs

Déclaration C (statique)

- Algo :
 nom : Tableau [1..taille] de type
- C :

```
type  nomTableau[ taille ];
```

- Les tableaux commencent à l'index 0
- Aucune variation de la taille pendant l'exécution
- Nécessite normalement une taille connue à l'avance

```
#define DIMENSION 3  
float  tflt_vecteur1 [DIMENSION];  
int    tint_vecteur2 [15];
```

- Stocké consécutivement en mémoire
- Il n'existe aucune fonction qui donne la taille d'un tableau

Utilisation

- `vecteur1 = vecteur2;` est illégal
- `vecteur1 + vecteur2;` est illégal
- `vecteur1 = 0;` est illégal
- `vecteur1` désigne le tableau
- `vecteur1[indice]` désigne la case d'index `indice`

À proscrire !

```
int int_taille;  
... scanf ("%d" , &int_taille );  
... / ...  
{  
    int tableau[int_taille];  
    ... / ...  
}
```

- Possibilité d'erreur

Attention

Passage de paramètre

```
.../...  
void toto (int int_nb) {  
    int_nb = 5;  
}  
.../...  
int main (int argc, char** argv) {  
    int int_nb;  
    int_nb = 4;  
    toto(int_nb);  
    printf("%d", int_nb); // Affiche 4  
    .../...  
}
```

Attention

Modification du tableau

```
.../...  
void toto (int tint_tab[N]) {  
    tint_tab[0] = 5;  
}  
.../...  
int titi (int tint_tab[N]) {  
    tint_tab[0] = 4;  
    printf("%d", tint_tab[0]); // Affiche 4  
    toto(tint_tab);  
    printf("%d", tint_tab[0]); // Affiche 5  
    .../...  
}
```

- Principe du traitement par effet de bord

Tableau en argument

```
.../...  
int main (int argc, char** argv) {  
    int tint_note[N]; // Declaration tableau de note  
  
    // Tri du tableau  
    // Notez l'appel du tableau  
    tri_tableau(tint_note);  
    // Ici, le tableau a ete modifie  
    .../...  
    return(0);  
}
```

- Appel en passant le nom du tableau
- Pour l'instant pas de fonction qui retourne un tableau

Exemple : Afficher un tableau

```
void afficherTableau (int tint_tab[N]) {  
    int i;  
    for (i=0; i<N; ++i) {  
        printf("%d_", tint_tab[i]);  
    }  
}  
.../...  
int main (int argc, char** argv) {  
    int tint_tab[N]; // Declaration tableau de note  
    .../...  
    // Affiche le tableau  
    afficherTableau(tint_tab);  
    .../...  
    return(0);  
}
```

Commentaires doxygen

```
/*!  
 * \fn void afficherTableau (int tint_tab[N])  
 * \param tint_tab Tableau a afficher  
 * \brief Affiche un tableau d'entier de N cases  
 */  
void afficherTableau (int tint_tab[N]);
```

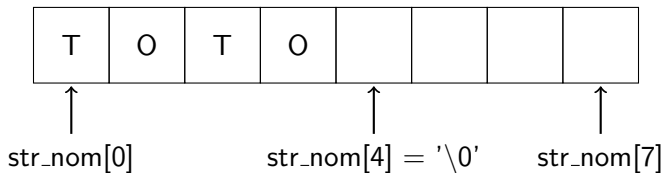
Il n'y pas la dimension du tableau dans la description du paramètre !

Cas particulier

Tableau de caractères

- `char nom[20];`
- Façon de déclarer des "chaînes" de caractères
- Manipulation délicate
- Permet de stocker $n - 1$ caractères
- Ne pas oublier le marqueur de fin de chaîne : `'\0'`
- Une case : un caractère

```
char str_nom[8];  
.../...
```



Attention

```
char str_nom[20];  
int int_retour;  
  
int_retour = scanf("%s", str_nom);
```

- Pas de & lors de la saisie pour une chaîne

Matrice

Matrices

Déclaration statique

- `int tint_mat[3][2];`
- Déclare un tableau de 3 lignes par 2 colonnes
- Taille fixe, même nombre de colonnes par ligne
- Choix arbitraire pour l'ordre des dimensions
- Nécessaire de garder une cohérence dans la notation
- Besoin de connaître la convention lors de la communication avec d'autres développeurs
- Stocké consécutivement en mémoire \Leftrightarrow possibilité de débordement et/ou recouvrement

Exemple

```
int tint_matrice[N][M];  
.../...  
for (i=0; i<N; ++i) {  
    for (j=0; j<M; ++j) {  
        tint_matrice[i][j] = rand()%20;  
    }  
}
```