

Travaux pratiques n°6 (2h): Les tableaux à une dimension

Objectifs

- Déclarer et utiliser des tableaux en langage C
- Approfondir l'utilisation d'itératives

Documents nécessaires

- sujet du TD6 et notes de TD6
- support de cours sur les tableaux

Partie 1: Compléments pratiques au langage C

Le préprocesseur du C:

Nous avons vu en cours que la directive *define* permet de déclarer des constantes:

```
#define MAX 10
```

```
#define AGECADET 12
```

```
#define AGEMINIME 10
```

```
#define AGEPUVILLE 8
```

```
#define AGEPOUSSIN 6
```

Ces directives ne sont pas des instructions, les constantes ainsi définies seront remplacé par leur valeur à la compilation par le pré-compilateur.

Ainsi considérons l'extrait de programme suivant:

```
if (age >= AGECADET)
    afficher("Catégorie cadet");
else
    if (age >= AGEMINIME)
        afficher ("Catégorie minime");
    else
        if (age >= AGEPUVILLE)
            afficher("Catégorie pupille");
```

```
else
    if (age >= AGEPOUSSIN)
        afficher("Catégorie poussin") ;
```

à la compilation le pré-compilateur remplacera les constantes par leur valeur avant la traduction ce qui donnera après pré-compilation et avant traduction:

```
if (age >= 12)
    afficher("Catégorie cadet") ;
else
    if (age >= 10)
        afficher ("Catégorie minime") ;
    else
        if (age >= 8)
            afficher("Catégorie pupille") ;
        else
            if (age >= 6)
                afficher("Catégorie poussin") ;
```

La directive permet ainsi de paramétrer son programme à moindre coût (pas de place réservée pour les constantes).

Pour les tableaux elle est utile pour définir les tailles:

```
#define MAX 10
...
int tab[MAX];
```

Pas de test de limite sur les tableaux en C:

Comme on vous l'a indiqué en cours, à l'exécution, il n'y a aucun test de dépassement de tableau. Pour vous en convaincre, tester le programme suivant:

```
#include <stdio.h>
int main(void)
{
    int tab1[10] = {1,2,3,4,5,6,7,8,9,10};
    int tab2[10] = {20,21,22,23,24,25,26,17,28,29};
```

```
int i;
for (i=0; i < 20; i++)
    printf("\n element %d de tab2 %d", i,tab2[i]);
return (0);
}
```

Question:

Qu'en déduisez vous?

Partie 2: Traduire des corps d'algorithmes détaillés avec tableaux en langage C

Exercice 1

Écrire un programme qui propose à l'utilisateur de faire les actions proposées dans le menu suivant:

- 0- Initialiser le tableau de notes
- 1- Afficher le tableau de notes
- 2- Calculer la moyenne des notes du tableau
- 3- Donner la meilleure note du tableau
- 4- Rechercher une note dans le tableau
- 5- Rechercher le nombre d'occurrences d'une note dans le tableau
- 6- Sortir

Exercice 2

- Écrire le programme correspondant à l'exercice 4 (recherche de séquence)
 - Écrire le programme correspondant à l'exercice 5 (palindrome de phrase).
-