

Le C - Complements

Programmation impérative

R.Gosswiller

1 #Include

2 Tableaux

#Include

#include

Principe

La fonction include permet de lister les autres fichiers qui seront utiles lors de la compilation : bibliothèques, fichiers d'en-tête (.h), etc.

Les instructions includes sont à placer **au début du fichier**.

Syntaxe

```
1 #include <stdio.h>      //consulte le repertoire de bibliotheques
2 #include "myFile.h"     //consulte le dossier courant
3 #include "C:\\monprojet\\myFile.h" //fichier specifique
```

A noter

On nomme module un ensemble de fichiers .c et .h qui constituent un ensemble cohérent à l'intérieur d'un projet.

Usages

Principe

Les inclusions sont principalement pour trois raisons :

- Inclure des bibliothèques externes typiques comme `<stdlib.h>`, `<stdio.h>` ou `<maths.h>`
- Eviter les problèmes de référencement internes à un fichier
- Lier les différents fichiers de code d'un projet

Exemple

file1.c

```
1  #include <stdlib.h>
2  #include <stdio.h>
3
4  int a = 0;
5
6  int multipleTrois(int a) {
7      a++;
8      int resultat = (a % 3 == 0) ? 1 : 0;
9      return resultat; }
10
11 void main() {
12     for(int i = 0; i<=10; i++)
13     {
14         if (multipleTrois(i) != 0)
15         {
16             printf("%d est multiple de trois\n", i);
17         }
18     }
19     return; }
```

Exemple

file1.h

```
1  #ifndef _FILE1HEADER
2  #define _FILE1HEADER
3
4  #include <stdlib.h>
5  #include <stdio.h>
6
7  int a;
8
9  int multipleTrois(int)
```

Tableaux

Tableaux

Principe

(le cours ne s'attardera pas sur les tableaux de manière générale. Pour plus de renseignements, cf le cours de python sur les structures de données !)

Tableaux en C

En C, la taille des tableaux doit toujours être indiquée, et les éléments doivent être initialisés.

Syntaxe : `typeElement nomVariable[nombreCases] = elements`

Une fois le nombre de cases choisi, il ne peut plus être modifié !

Syntaxe

```
1  int monTableau[4] = {0, 10, 5, 3}
2  int monTableau[1000] = {0} // tous les elements a 0
```

Tableaux à deux dimensions (voire plus)

Principe

Tout tableau en C peut être défini comme une matrice

Tableau de tableaux

Cases numérotées en n dimensions

Rôle

Composition d'informations

Gestion multiple de données

Tableaux de chaînes (ex : `argv`)

Tableaux à deux dimensions

Utilisation multiple de l'opérateur []

Syntaxe

```
1  int tab[10][20];
2  tab[3][4] = value;
3
4  char tab2[10][20];
5  tab2[0] = "hello";
6  tab2[1] = "world";
7
8  printf("%c", tab[1][1]); // Affiche o
9  printf("%s", tab[1]); // Affiche world
```

Conclusion

- `#include` permet d'utiliser plus qu'un seul fichier
- Les tableaux doivent être définis en type et en taille