

TP2 – Syntaxe de base du C++

Exercice 1 : allocation statique / allocation dynamique

Déclarer un pointeur vers un entier puis construire cet entier et l'initialiser avec une valeur entière. Afficher la valeur de l'entier et l'adresse de son emplacement mémoire.

Refaire la même chose avec un réel.

Exercice 2 : passage par valeur / passage par référence / passage par adresse

Ecrire trois version d'une fonction qui teste l'égalité entre deux entiers (utiliser le passage par valeur, par adresse et par référence). Tester cette fonction dans main.

Exercice 3 : références et pointeurs

Écrire une fonction nommée *incrémenter()* permettant d'incrémenter la valeur d'une variable passée en paramètre et une fonction nommée *permuter()* permettant d'échanger les contenus de 2 variables de type int fournies en argument :

- en transmettant l'adresse des variables concernées (seule méthode utilisable en C) ;
- en utilisant la transmission par référence.

Dans les deux cas, écrire un programme (main) qui teste les deux fonctions.

Exercice 4 : fonctions / passage par références

Ecrire un programme en C qui permet de simuler le fonctionnement d'une calculatrice :

```
***** CALCULATRICE *****
* 1. Addition                **
* 2. Multiplication          **
* 3. Soustraction            **
* 4. Division                **
* 5. Quitter                 **
*****
Operation?
```

1. L'utilisateur choisira l'opération à effectuer et saisira les opérandes. Le programme affichera le résultat de l'opération.
2. Le programme effacera la console et affichera le menu de nouveau tant que l'utilisateur n'a pas choisi de quitter.

Pour effacer la console on peut utiliser l'instruction **system("cls")**.

Exercice 5 : les nombres aléatoires

Ecrire une fonction qui permet de générer des réels (resp. des entiers) aléatoires dans un intervalle donné. Dans la fonction main faire l'appel 5 fois à cette fonction et afficher les nombres aléatoires générés. Utiliser la bibliothèque « random ».

Exercice 6 :

Ecrire un programme permettant de dessiner un sapin en fixant la hauteur du cône, la largeur et la hauteur du tronc et le nombre de boules (dont la position est aléatoire). Par exemple, pour un cône de hauteur 6, un tronc de hauteur 2 et de largeur 3 et 4 boules on doit obtenir un affichage du type :

```
      *
    *O*
  * * * * *
 *O* *O*
* * * * *O* * * *
* * * * *O* * * *
* * * * *O* * * *
      | | |
      | | |
```

Pour résoudre le problème, il est judicieux de le décomposer en trois étapes :

- Dessiner le cône du sapin qui dépend de sa hauteur. Soit N la hauteur du cône, il s'agit d'imprimer à l'écran N lignes dont la première est composée d'une étoile et la dernière de $2 * N - 1$ étoiles (voir dessin).
- Rajouter le tronc.
- Modifier le dessin du cône en rajoutant des boules de manière aléatoire.

Suggestion pour l'étape 3 : Avant d'afficher une étoile, générer un nombre aléatoire compris entre

0 et 1. Si ce nombre est inférieur à 0,5 (probabilité de placer une boule dans notre exercice) et s'il reste des boules à placer, afficher un O sinon afficher une étoile.

Exercice 7 : fonction récursive

Écrire un programme qui demande à l'utilisateur de taper un entier N et qui calcule $u(N)$ défini par :

$$u(0)=3 ;$$

$$u(n+1)=3.u(n)+4.$$