

C++, Feuille de TP n° 2 : Conditionnelles

Remarque globale : Chaque question doit être testée avant de passer à la suivante avec un jeu d'essais le plus complet possible. Ajouter en commentaire en fin de programme, la liste des différents tests effectués, avec une explication du choix de chaque test (une ligne par test).

Vous référer au poly C++ Volume 1 pour les détails de la syntaxe.

Exercice 1

Ecrire un programme demande à l'utilisateur qui lit deux valeurs réelles **a** et **b** puis qui affiche sur le flot de sortie « ordre croissant » si a est plus petit ou égal à b, et « ordre décroissant » si b est plus petit que a. Bien tester tous les cas de figure (ex: 1 et 5, 5 et 1, 5 et 5), et les noter en commentaire.

Exercice 2

Ecrire un programme qui lit un nombre entier et affiche s'il est pair ou non.

Rappel : un nombre entier n est pair si sa valeur modulo 2 - reste de la division par 2 - qui s'écrit $n\%2$ en C++ est nulle. Attention à l'opérateur de comparaison en C++.

Exercice 3

Complétez le programme suivant pour qu'il lise trois nombres x, y et z, affecte à min la plus petite des 3 valeurs, et à max la plus grande, puis affiche le résultat.

```
int main()
{
    double min, max;
    double x, y, z;
    ...
}
```

Exercice 4

Copiez le fichier `lireNombre.cpp` du répertoire Dokeos dans votre répertoire de projet.

Indentez le programme de façon habituelle pour que l'identification des blocs soit possible à l'oeil nu.

Compilez. Corrigez les erreurs. Quels résultats affiche le programme lorsqu'on lui fournit en donnée les valeurs suivantes : a. 0, b. 1, c. 4, d. 10, e. -5

Exercice 5

En utilisant l'instruction switch, écrire un programme qui lit un caractère ('p' pour printemps, 'e' pour été, 'a' pour automne, 'h' pour hiver) et affiche le nom complet de la saison correspondante. Le programme affichera sur le flot de sortie un message d'erreur si la lettre tapée ne correspond pas à une saison.

Exercice 6 à rendre commenté accompagné du tableau du jeu d'essais

Écrire un programme affiche le prix à payer en fonction de la grille de tarifs indiquée dans la table qui suit. Le programme commence par demander l'âge de l'individu, s'il est abonné ou non ('o' ou 'n'), et le type de spectacle ('court ou long : c' ou 'l'). Pas de vérification dans un premier temps. Il calcule et affiche ensuite le prix à payer. Il est conseillé d'utiliser une variable prix. Attention à ne pas tester 2 fois la même condition.

	Spectacle court		Spectacle long	
	abonné	non abonné	abonné	non abonné
Enfant (<18 ans)	6	8	10	12
Adulte	15	18	18	22

Complétez le programme en effectuant une vérification sur la saisie de l'abonnement (si l'utilisateur n'entre pas 'o' ou 'n', le programme s'arrête après avoir affiché la raison de l'arrêt). NB : l'instruction `exit(0)` provoque

l'arrêt d'un programme. Elle nécessite l'inclusion suivante en début de fichier

Exercice 7 (*) à rendre commenté accompagné du tableau du jeu d'essais

Ordinaux abrégés en anglais

Dans cet exercice, nous nous intéressons aux ordinaux abrégés, où le nombre est écrit en chiffres.

Les premiers sont 1st », « 2nd », « 3rd », « 4th », etc. (abréviations de « first, second, third, fourth, ... »).

Pour déterminer le suffixe, on regarde le dernier chiffre du nombre : si c'est 1, on ajoute le suffixe est -st ; si c'est 2, le suffixe est -nd ; si c'est 3, le suffixe est -rd ; sinon le suffixe est -th.

Il y a une exception : si l'avant-dernier chiffre du nombre est 1, le suffixe est toujours -th.

Ecrire un programme qui lit un nombre, et qui affiche l'ordinal anglais abrégé correspondant.

Rappel : le dernier chiffre d'un nombre `val` vaut exactement `val%10`