

Introduction à l'informatique

TP2 - conditionnelles, boucles

Exercice 1 (Instructions conditionnelles).

Écrire un programme qui demande l'âge d'un enfant à l'utilisateur. Ensuite, il l'informe de sa catégorie :

- Rien si inférieur (strict) à 6 ans
- "Poussin" de 6 à 7 ans
- "Pupille" de 8 à 9 ans
- "Minime" de 10 à 11 ans
- "Cadet" de 12 à 13 ans
- "Junior" de 14 à 17 ans
- "Senior" à partir de 18 ans

Exercice 2 (Boucles simples).

Écrivez un programme qui affiche les entiers de 1 à 20 par pas de 1 séparés par une virgule et un espace en utilisant une boucle for. Vous devez obtenir le résultat ci-dessous :

```
Bonjour! Je vais compter de 1 jusqu'a 20 :  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20,
```

Si vous obtenez le résultat ci-dessus avec une virgule à la fin, modifiez le programme pour avoir le résultat suivant :

```
Re-Bonjour! Je vais compter de 1 jusqu'a 20 :  
1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20
```

Exercice 3 (Boucles simples).

Écrivez un programme qui calcule la somme des nombres de 1 à 10, et affiche toutes les sommes intermédiaires puis la somme finale. L'affichage désiré ressemble au suivant :

```
Somme intermediaire numero 0 est 0,  
Somme intermediaire numero 1 est 1,  
Somme intermediaire numero 2 est 3,  
Somme intermediaire numero 3 est 6,  
Somme intermediaire numero 4 est 10,  
Somme intermediaire numero 5 est 15,  
Somme intermediaire numero 6 est 21,  
Somme intermediaire numero 7 est 28,  
Somme intermediaire numero 8 est 36,  
Somme intermediaire numero 9 est 45,  
Somme intermediaire numero 10 est 55,  
  
Somme des nombres de 1 a 10 : 55
```

Exercice 4 (Boucles `while (cond) {... } , do {... } while (cond)`).

- (1) Écrivez un programme qui indique si la valeur d'une variable saisie au clavier est paire ou impaire.
- (2) Modifiez le programme précédent pour qu'il permette de lire plusieurs valeurs à la suite et afficher la parité de chaque nombre saisi. L'utilisateur saisit la valeur 0 pour indiquer la fin de l'entrée des valeurs. Pour cela, on utilisera une boucle `do ... while`.

Votre programme précédent indique-t-il la parité de la valeur 0 qui devait juste indiquer la fin des saisies ? Si c'est le cas, modifiez-le pour que 0 ne soit plus considéré comme une valeur à traiter !

- (3) À la suite de la boucle `do ... while` précédente faites une boucle `while` qui ait la même fonction que précédemment.

Exercice 5 (Instructions conditionnelles et boucles simples).

- (1) Écrire un programme qui demande à l'utilisateur d'entrer un nombre positif ou nul. Ce programme devra afficher un message d'erreur et demander à l'utilisateur de saisir un nouvel entier tant que le nombre entré au clavier est strictement négatif.
- (2) Écrire un programme qui demande à l'utilisateur d'entrer un nombre inscrit entre une borne supérieure et une borne inférieure tel que :

`borneMin <= nb <= borneMax`

Si la borne inférieure est supérieure à la borne supérieure, le programme devra commencer par les échanger pour que `borneMin` soit toujours inférieure ou égale à `borneMax`. Un message pourra être écrit pour en informer l'utilisateur.

Ce programme devra afficher un message d'erreur et demander à l'utilisateur de saisir un nouvel entier tant que le nombre entrée n'est pas compris entre `borneMin` et `borneMax`.

Exercice 6 (Puissance de 3).

Vous devez vérifier si un nombre est une puissance de trois et afficher le résultat/ De plus, si c'est le cas, vous devez afficher l'exposant qui va avec. Le nombre en question est stocké dans une variable nommée `number` de type `int`.

Attention, si le reste du modulo de 3 vaut 0, le nombre n'est pas forcément une puissance de 3. Par exemple :

$6 \% 3 = 0$ et 6 n'est pas une puissance de 3.

Pour savoir si un nombre est une puissance de trois, vous pouvez plutôt calculer combien de fois le nombre est divisible par 3. Ainsi vous vous posez à chaque fois la question de savoir si votre nombre est multiple de 3, et espérez arriver jusqu'à 1.

Exemples :

$6 / 3 = 2$	pas divisible par 3	→ 6 n'est pas une puissance de 3
$9 / 3 = 3$	et $3 / 3 = 1$	→ 9 est une puissance de 3
$18 / 3 = 6$	et $6 / 3 = 2$	pas divisible par 3 → 18 n'est pas une puissance de 3
$45 / 3 = 15$	et $15 / 3 = 5$	pas divisible par 3 → 45 n'est pas une puissance de 3
$27 / 3 = 9$	et $9 / 3 = 3$ et $3 / 3 = 1$	→ 27 est une puissance de 3

Exercice 7 (Boucle).

Écrivez un programme qui affiche la table de multiplication d'un nombre saisi au clavier, le nombre de multiples de ce nombre que l'on souhaite voir est aussi saisi par l'utilisateur au clavier.

Vous devez obtenir le résultat suivant :

```
Saisissez le nombre dont vous voulez afficher la table de multiplication : 9
Combien de multiple voulez vous afficher ? 12

La table de multiplication du chiffre 9 que vous souhaitez est :
9 * 1 = 9
9 * 2 = 18
9 * 3 = 27
9 * 4 = 36
9 * 5 = 45
9 * 6 = 54
9 * 7 = 63
9 * 8 = 72
9 * 9 = 81
9 * 10 = 90
9 * 11 = 99
9 * 12 = 108
```

Exercice 8 (Doubles boucles).

Écrire un programme qui trace un rectangle dont le bord est formé d'étoiles. On demandera la largeur et la hauteur à l'utilisateur (valeur de 0 à 256). On fera en sorte qu'après chaque affichage l'utilisateur en entrant "1" pourra retracer un nouveau rectangle et s'il entre 2 on sortira du programme. Penser aux cas particuliers largeur = 1 et/ou hauteur = 1. On obtiendra le résultat suivant :

```
Entrer la largeur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 6
Entrer la hauteur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 4

*****
*   *
*   *
*   *
*****

Faites 1 pour recommencer et 2 pour quitter : Veuillez saisir un entier entre 1 et 2 : 1
Entrer la largeur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 1
Entrer la hauteur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 1

*

Faites 1 pour recommencer et 2 pour quitter : Veuillez saisir un entier entre 1 et 2 : 1
Entrer la largeur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 1
Entrer la hauteur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 2

*
*

Faites 1 pour recommencer et 2 pour quitter : Veuillez saisir un entier entre 1 et 2 : 1
Entrer la largeur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 2
Entrer la hauteur du Rectangle : Veuillez saisir un entier entre 0 et 256 : 1

**

Faites 1 pour recommencer et 2 pour quitter : Veuillez saisir un entier entre 1 et 2 : 2
```

Exercice 9 (Triangle d'étoiles).

Écrire un programme qui affiche un triangle rempli d'étoiles (*) sur un nombre de lignes saisi par l'utilisateur. Votre programme doit demander à l'utilisateur s'il veut continuer ('o' ou 'O') ou arrêter ('n' ou 'N'). Vous devriez obtenir la trace suivante :

```

Veillez saisir un entier entre 1 et 100 : 3
*
**
***

Voulez-vous continuer ('o' ou 'O' pour oui et 'n' ou 'N' pour non) ? p
ERREUR : Merci d'entrer 'o' ou 'O' ou 'n' ou 'N' et pas un autre caractere

Voulez-vous continuer ('o' ou 'O' pour oui et 'n' ou 'N' pour non) ? o
Veillez saisir un entier entre 1 et 100 : 5
*
**
***
****
*****

Voulez-vous continuer ('o' ou 'O' pour oui et 'n' ou 'N' pour non) ? n

```

Exercice 10 (Suite de Fibonacci).

On souhaite calculer les termes de la suite de Fibonacci jusqu'au rang n. la valeur de n est demandée a l'utilisateur au clavier. La suite de Fibonacci est telle que :

$$U_0 = 1 \quad \text{et} \quad U_1 = 1 \quad \text{et} \quad U_n = U_{n-1} + U_{n-2}$$

On obtiendra l'affichage suivant :

```

Quel est le rang max du terme de la suite de Fibonacci a calculer >= 2 et <= 50 :
Veillez saisir un entier entre 2 et 50 : 22
U0 : 1
U1 : 1
U2 : 2
U3 : 3
U4 : 5
U5 : 8
U6 : 13
U7 : 21
U8 : 34
U9 : 55
U10 : 89
U11 : 144
U12 : 233
U13 : 377
U14 : 610
U15 : 987
U16 : 1597
U17 : 2584
U18 : 4181
U19 : 6765
U20 : 10946
U21 : 17711
U22 : 28657

```

EXERCICES SUPPLEMENTAIRES

Exercice 11 (Calcul de la valeur approchée de PI).

Calculer la valeur de PI à epsilon près par la formule d'EULER

$$\text{Pi}/4 = 1 - 1/3 + 1/5 - 1/7 + 1/9 - 1/11 + 1/13 - 1/15 \dots$$

La valeur d'épsilon sera entrée au clavier par l'utilisateur (exemple : 0.00001).

Pour choisir le nombre de chiffres affichés après le virgule pour un nombre réel, on pourra utiliser l'instruction suivante : `cout << setprecision(10) << reel ;`

Remarquez que pour utiliser `setprecision`, on a dû ajouter `#include<iomanip>` dans l'en-tête.

Exercice 12 (Chiffres).

Écrire un programme qui demande d'entrer un entier positif au clavier (pour s'assurer que l'entier est positif on pourra utiliser une boucle `while`). Le programme calculera et affichera le nombre de chiffres qui compose ce nombre. Il calculera aussi et affichera le plus grand des chiffres, le plus grand chiffre pair et le plus grand des chiffres impair :

```
Entrer un entier 32 bits (max 4294967295) au clavier et 0 pour sortir : 35729
Le nombre de chiffre est : 5
Le plus grand des chiffres est : 9
Le plus grand des chiffres pair est : 2
Le plus grand des chiffres impair est : 9
Entrer un entier 32 bits (max 4294967295) au clavier et 0 pour sortir : 45628
Le nombre de chiffre est : 5
Le plus grand des chiffres est : 8
Le plus grand des chiffres pair est : 8
Le plus grand des chiffres impair est : 5
Entrer un entier 32 bits (max 4294967295) au clavier et 0 pour sortir : 0
Le nombre de chiffre est : 1
Le plus grand des chiffres est : 0
Le plus grand des chiffres pair est : 0
Le nombre n'a pas de chiffre impair
```

Exercice ♠ 13 (Marches avec boucles `for`).

Écrire le programme qui affiche la marche d'une personne qui monte dans un immeuble de n étages jusqu'au sommet avec un nombre fixe de marche par étage. Le nombre de marches et d'étage est entré au clavier. Utilisez deux boucles `for`. Cela devrait ressembler à :

```
Donnez le nombre d'etages : 8
Donnez le nombre de marches par etage : 18

    Vous etes au rez de chaussee
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 1 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 2 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 3 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 4 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 5 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 6 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 7 .....
-----
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
    Vous avez atteint l'etage 8 .....
-----
Vous avez monte au total 144 marches.
```

Exercice ♠ 14 (Marches avec boucles `while`).

Reprenez l'exercice précédent et modifiez le pour qu'il effectue le même traitement, mais avec des boucles `while`.

Exercice 15 (Suite numérique).

On peut approcher la valeur de la racine d'un entier a à l'aide d'une simple suite numérique, définie telle que : $U_0 = 0$, $U_{n+1} = U_n + a / U_n + 1$

Écrire une programme qui calcule la racine d'un nombre entier saisi par l'utilisateur en utilisant cette méthode :

```

Entrer le nombre dont l'on veut la racine : 10
Entrer le nombre de termes de la suite : 30

***** calcul de la racine de 10*****
10
1.81818187236786
4.19354820251465
2.73291921615601
3.41098189353943
3.0403618812561
3.2275230884552
3.12890625
3.17975401878357
3.15323662757874
3.16698479652405
3.15983510017395
3.16354751586914
3.16161823272705
3.16262030601501
3.16209959983826
3.16236996650696
3.16222977638245
3.16230273246765
3.162264585495
3.16228437423706
3.16227436065674
3.1622793674469
3.16227698326111
3.16227793693542
3.16227746009827
3.16227793693542
3.16227746009827
3.16227793693542

```

Exercice 16 (Instructions conditionnelles).

Soit une famille composée d'un père d'une mère et de 0 à 3 enfants. On fera l'acquisition de l'âge du père, de la mère et des enfants (de -1 à 99 ans) :

-1 signifie que cet enfant n'existe pas et 0 que l'enfant est né et n'a pas encore 1 an.

Bien sûr si le premier enfant n'existe pas on ne demandera pas l'âge de 2ème et du 3ème enfant, de même si le 2ème enfant n'existe pas on ne demande pas celui du 3ème enfant.

L'âge du second et du 3ème enfant doit être égal ou inférieur à celui du premier et l'âge du 3ème égal ou inférieur à l'âge du second (cas de jumeaux ou de triplets).

Si l'on sait ce qu'est une fonction, on pourra créer et utiliser une fonction `int lireEntierEntre (int min, int max)` qui permettra de lire un entier n au clavier tel que $\min \leq n \leq \max$. cette fonction retournera l'entier n lu au clavier.

On suppose qu'une jeune fille ne peut pas être mère avant l'âge de 12 ans et un garçon ne peut pas être père avant 13 ans.

- (1) On vérifiera cette condition et si elle n'est pas vérifiée on affichera un message d'erreur et on sortira du programme.
- (2) Si la condition est vérifiée on va calculer la réduction accordée à cette famille par la SNCF :
 - Si la famille n'a pas d'enfant elle n'a pas de réduction.

- Si la famille à 1 enfant elle a droit à une réduction de 5%.
- Si la famille à 2 enfants elle a droit à une réduction de 15%.
- Si la famille à 3 enfants elle a droit à une réduction minimale de 30% mais elle a droit à une réduction plus grande si :
 - Les 3 enfants ont moins de 12 ans la réduction est de 35%.
 - Si 2 enfants ont moins de 12 ans et le 3ème est âgé de 12 à 18 ans, la réduction est de 38%.
 - Si 1 enfant a moins de 12 ans et les 2 autres sont âgés de 12 à 18 ans, la réduction est de 42%.
 - Si les 3 enfants sont entre 12 et 18 ans, la réduction est de 50%.
 - Si l'un des parents au moins est âgé de 65 ans ou plus une réduction de 5% est accordée en plus.

Écrire le programme qui calculera la réduction accordée et affichera par exemple :

Exercice ♠ 17 (Pyramide).

Écrivez un algorithme qui affiche cette pyramide de nombres en utilisant des boucles. Le nombre de lignes sera demandé à l'utilisateur.

```

0
121
23432
3456543
456787654
56789098765
6789012109876
789012343210987
89012345654321098
9012345678765432109
012345678909876543210
12345678901210987654321
2345678901234321098765432
345678901234565432109876543
45678901234567876543210987654
5678901234567890987654321098765
678901234567890121098765432109876
78901234567890123432109876543210987
8901234567890123456543210987654321098
901234567890123456787654321098765432109
01234567890123456789098765432109876543210
1234567890123456789012109876543210987654321
234567890123456789012343210987654321098765432
34567890123456789012345654321098765432109876543
4567890123456789012345678765432109876543210987654
567890123456789012345678909876543210987654321098765
67890123456789012345678901210987654321098765432109876
7890123456789012345678901234321098765432109876543210987
890123456789012345678901234565432109876543210987654321098
90123456789012345678901234567876543210987654321098765432109

```

On pourra faire ensuite une pyramide de lettre minuscule de a à z.