

TD 2 : Conditions et boucles

Objectif : S'initier à la programmation en utilisant les boucles et conditions

Exercice 1 : Ecrire un algorithme demandant à l'utilisateur de saisir deux nombre et d'afficher le plus grand des deux

Exercice 2 : Ecrire un algorithme demandant à l'utilisateur de saisir trois nombres et d'afficher le plus grand des trois

Exercice 3 : Dans un programme de calcul d'une facture, on veut effectuer une remise de 1% si le montant de la facture dépasse 1000 frs. Ecrire un algorithme demandant à l'utilisateur de saisir la quantité du produit qu'il désire acheter et son prix unitaire. Ensuite on calcule le montant de la facture. Si l'utilisateur bénéficie d'une remise, tenez le informer.

Exercice 4 : Ecrire un algorithme demandant à l'utilisateur de saisir les données suivantes :

- Le montant HT
- Le nombre d'article
- La TVA

Puis de calculer le montant TTC selon la règle suivante :

- Remise = 10% si le nombre d'article > 10
- Remise = 0.75% si $5 < \text{nombre d'article} \leq 10$
- Remise = 0.5% si nombre d'article ≤ 5

Exercice 6 : Ecrire un algorithme permettant de résoudre une équation du 2nd degré sous la forme $ax^2 + bx + c = 0$. Au préalable, l'utilisateur doit saisir les valeurs des coefficients a, b et c qui sont des nombres réels

Exercice 7 : Ecrire un algorithme demandant à l'utilisateur de saisir la note comprise entre 0 et 20 et affichant les mentions suivantes :

- « Faible » si note < 10
- « A. bien » si $10 \leq \text{note} \leq 12$
- « bien » si $14 \leq \text{note} \leq 16$
- « T. bien » si $16 \leq \text{note} \leq 18$
- « Excellent » si note > 18

Exercice 8 : Écrivez un programme pour saisir un caractère de l'utilisateur et vérifiez si le caractère donné est un alphabet, un chiffre ou un caractère spécial en utilisant if-else.

- Un caractère est un alphabet s'il se situe entre a-z ou A-Z.
- Un caractère est un chiffre s'il est compris entre 0 et 9.
- Un caractère est un symbole spécial s'il est ni alphabet ni chiffre.

Exercice 8bis : Ecrire un code permettant de calculer la somme des 100 premiers entiers naturels et d'afficher le résultat.

Exercice 9 : Ecrire un algorithme demandant à l'utilisateur de saisir un nombre entier (qu'on appellera n) puis de calculer la somme des n nombre impair et d'afficher le résultat.

Exercice 10 : Ecrire un algorithme qui demande à l'utilisateur un nombre N, puis calcule la

somme des nombres de 0 à N

Exemples

N = 6

Somme = $0+1+2+3+4+5+6 = 21$

Exercice 11 : Ecrire un algorithme qui demande un nombre de départ, et qui ensuite affiche les dix nombres suivants. Par exemple, si l'utilisateur entre le nombre 17, le programme affichera les nombres de 18 à 27.

Exercice 12 : Ecrire un code demandant à l'utilisateur de saisir un entier n puis calcule son factoriel ($8! = 8*7*6*5*4*3*2*1$)

Exercice 13 : Ecrire un algorithme qui demande successivement 20 nombres à l'utilisateur, et qui lui dise ensuite quel était le plus grand parmi ces 20 nombres :

Entrez le nombre numéro 1 : 12

Entrez le nombre numéro 2 : 14

etc.

Entrez le nombre numéro 20 : 6

Le plus grand de ces nombres est : 14

Modifiez ensuite l'algorithme pour que le programme affiche de surcroît en quelle position avait été saisie ce nombre :

C'était le nombre numéro 2

Exercice 14 : Ecrire un algorithme demandant à l'utilisateur de saisir un nombre entre 1 et 10. Ensuite, il affichera la table de multiplication du nombre saisi. Attention, si le nombre saisi par l'utilisateur n'est pas compris entre 1 et 10, tenez le informer et demander lui de recommencer à nouveau

Exercice 15 : Reprendre l'exercice 7 le mais en imposant une note comprise entre 0 et 20. Si l'utilisateur saisit une note en dehors de 0 et 20, tenez le informer et demander lui de recommencer à nouveau

Exercice 16 : On désire sécuriser une enceinte pressurisée. On se fixe une pression seuil et un volume seuil : $p_{\text{Seuil}} = 2.3$, $v_{\text{Seuil}} = 7.41$.

On demande de saisir la pression et le volume courant de l'enceinte et d'écrire un code qui simule le comportement suivant :

- Si le volume et la pression sont supérieurs aux seuils : arrêt immédiat ;
- Si seule la pression est supérieure à la pression seuil : demander d'augmenter le volume de l'enceinte ;
- Si seul le volume est supérieur au volume seuil : demander de diminuer le volume de l'enceinte ;
- Sinon déclarer que « tout va bien ».

Ecrire ce programme.

Exercice 17 :

On souhaite réaliser un jeu s'appelant « Plus ou moins ». Le but du jeu est de trouver un nombre mystère (nombre tiré au hasard par l'ordinateur) en un minimum de coups.

Le principe est le suivant.

- L'ordinateur tire au sort un nombre entre 1 et 100.

- Il vous demande de deviner le nombre. Vous entrez donc un nombre entre 1 et 100.
- L'ordinateur compare le nombre que vous avez entré avec le nombre « mystère » qu'il a tiré au sort. Il vous dit si le nombre mystère est supérieur ou inférieur à celui que vous avez entré.
- Puis l'ordinateur vous redemande le nombre.
- ... Et il vous indique si le nombre mystère est supérieur ou inférieur.
- Et ainsi de suite, jusqu'à ce que vous trouviez le nombre mystère.

Exemple :

Quel est le nombre ? 50
 C'est plus !
 Quel est le nombre ? 75
 C'est plus !
 Quel est le nombre ? 85
 C'est moins !
 Quel est le nombre ? 80
 C'est moins !
 Quel est le nombre ? 78
 C'est plus !
 Quel est le nombre ? 79
 Bravo, vous avez trouvé le nombre mystère en 5 temps.
 Quel est le nombre ? 50
 C'est plus !
 Quel est le nombre ? 75
 C'est plus !
 Quel est le nombre ? 85
 C'est moins !
 Quel est le nombre ? 80
 C'est moins !
 Quel est le nombre ? 78
 C'est plus !
 Quel est le nombre ? 79
 Bravo, vous avez trouvé le nombre mystère en 5 temps.

Indication : Pour ce exercice, il faut nous falloir générer de façon aléatoire le mot mystère et mettre en place un gestionnaire de temps afin d'estimer le temps écoulé.

- Pour tirer un nombre aléatoire entre 1 et 100, nous allons utiliser la fonction « rand() » comme suit :

```
int MAX = 100, MIN = 1;
srand(time(NULL));
nombreMystere = (rand() % (MAX - MIN + 1)) + MIN;
int MAX = 100, MIN = 1;
srand(time(NULL));
nombreMystere = (rand() % (MAX - MIN + 1)) + MIN;
```

Il faudra pour cela inclure les librairies suivantes :

```
#include <stdio.h>
```

```
#include <stdlib.h>
#include <time.h>
#include <stdio.h>
#include <stdlib.h>
#include <time.h>
```

- Le temps écoulé correspond au nombre de fois qu'il aura mis pour trouver la réponse.