

EXERCICES – ALGORITHME SECONDE

Exercice 5.1

Ecrire un algorithme qui demande à l'utilisateur un nombre compris entre 1 et 3 jusqu'à ce que la réponse convienne.

[corrigé](#) - [retour au cours](#)

Exercice 5.2

Ecrire un algorithme qui demande un nombre compris entre 10 et 20, jusqu'à ce que la réponse convienne. En cas de réponse supérieure à 20, on fera apparaître un message : « Plus petit ! », et inversement, « Plus grand ! » si le nombre est inférieur à 10.

[corrigé](#) - [retour au cours](#)

Exercice 5.3

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.

[corrigé](#) - [retour au cours](#)

Exercice 5.4

Réécrire l'algorithme précédent, en utilisant cette fois l'instruction **Pour**

[corrigé](#) - [retour au cours](#)

Exercice 5.5

Ecrire un algorithme qui demande un nombre de départ, et qui ensuite écrit la table de multiplication de ce nombre, présentée comme suit (cas où l'utilisateur entre le nombre 7) :

Table de 7 :

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$

[corrigé](#) - [retour au cours](#)

Exercice 5.6

Ecrire un algorithme qui demande un nombre de départ, et qui calcule la somme des entiers jusqu'à ce nombre. Par exemple, si l'on entre 5, le programme doit calculer :

$$1 + 2 + 3 + 4 + 5 = 15$$

NB : on souhaite afficher uniquement le résultat, pas la décomposition du calcul.

[corrigé](#) - [retour au cours](#)

Exercice 5.7

Ecrire un algorithme qui demande un nombre de départ, et qui calcule sa factorielle.

NB : la factorielle de 8, notée 8 !, vaut

$$1 \times 2 \times 3 \times 4 \times 5 \times 6 \times 7 \times 8$$

[corrigé](#) - [retour au cours](#)

Exercice 5.8

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

[corrigé](#) - [retour au cours](#)

Exercice 5.9

Réécrire l'algorithme précédent, mais cette fois-ci on ne connaît pas d'avance combien l'utilisateur souhaite saisir de nombres. La saisie des nombres s'arrête lorsque l'utilisateur entre un zéro.

[corrigé](#) - [retour au cours](#)

Exercice 5.10

Lire la suite des prix (en euros entiers et terminée par zéro) des achats d'un client. Calculer la somme qu'il doit, lire la somme qu'il paye, et simuler la remise de la monnaie en affichant les textes "10 Euros", "5 Euros" et "1 Euro" autant de fois qu'il y a de coupures de chaque sorte à rendre.

[corrigé](#) - [retour au cours](#)

Exercice 5.11

Écrire un algorithme qui permette de connaître ses chances de gagner au tiercé, quarté, quinté et autres impôts volontaires.

On demande à l'utilisateur le nombre de chevaux partants, et le nombre de chevaux joués. Les deux messages affichés devront être :

Dans l'ordre : une chance sur X de gagner

Dans le désordre : une chance sur Y de gagner

X et Y nous sont donnés par la formule suivante, si n est le nombre de chevaux partants et p le nombre de chevaux joués (on rappelle que le signe ! signifie "factorielle", comme dans l'exercice 5.6 ci-dessus) :

$$X = n ! / (n - p) !$$

$$Y = n ! / (p ! * (n - p) !)$$

NB : cet algorithme peut être écrit d'une manière simple, mais relativement peu performante.

Ses performances peuvent être singulièrement augmentées par une petite astuce. Vous commencerez par écrire la manière la plus simple, puis vous identifierez le problème, et écrirez une deuxième version permettant de le résoudre.

[corrigé](#) - [retour au cours](#)

Corrigés des Exercices

Exercice 5.1

Variable N en Entier

Debut

$N \leftarrow 0$

Ecrire "Entrez un nombre entre 1 et 3"

TantQue $N < 1$ ou $N > 3$

Lire N

Si $N < 1$ ou $N > 3$ **Alors**