

Révisions algorithmiques

Exercice 5.1

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. Ecrire cet algorithme en utilisant une boucle POUR, puis en utilisant une boucle TANT QUE.

Exercice 5.2

Soit la fonction *random(m : entier, n : entier) : entier* renvoyant un nombre généré aléatoirement compris entre *m* et *n*.

Ecrire un algorithme qui choisit aléatoirement un nombre compris entre 0 et 255 à deviner puis demande à l'utilisateur un nombre compris entre 0 et 255 jusqu'à ce que la réponse convienne. A chaque tour de boucle, si la réponse fournie n'est pas bonne, l'algorithme fourni une indication à l'utilisateur : « c'est plus » ou « c'est moins »

Exercice 5.3

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 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
...  
7 x 10 = 70
```

Exercice 5.4

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.

Exercice 5.5

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

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

```
8 x 7 x 6 x 5 x 4 x 3 x 2 x 1 = 40320
```

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 saisi ce nombre :

```
C'était le nombre numéro 2
```

Exercice 5.9

Lire la suite des prix des achats d'un client (les prix sont en euros entiers). 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.

Exercice 5.10

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 écrivez une deuxième version permettant de le résoudre.