**Exercices Algorithmie**

**Partie I) Les Variables**

**Exercice 1.1)**

Quelles seront les valeurs des variables A et B après exécution des instructions suivantes ?

Variables A, B en Entier  
Début  
A ← 5  
B ← 2  
A ← B  
B ← A  
Fin

**Exercice 1.2)**

Plus difficile, mais c’est un classique absolu, qu’il faut absolument maîtriser : écrire un algorithme permettant d’échanger les valeurs de deux variables A et B, et ce quel que soit leur contenu préalable.

**Exercice 1.3)**

Une variante du précédent : on dispose de trois variables A, B et C. Écrivez un algorithme transférant à B la valeur de A, à C la valeur de B et à A la valeur de C (toujours quels que soient les contenus préalables de ces variables).

**Partie II) Affichage et Clavier**

**Exercice 2.1)**

Écrire un programme qui demande un nombre à l’utilisateur, puis qui calcule et  affiche le carré de ce nombre.

**Exercice 2.2)**

Écrire un programme qui lit le prix HT d’un article, le nombre d’articles et le taux de TVA, et qui fournit le prix total TTC correspondant. Faire en sorte que des libellés apparaissent clairement.

**Partie III) Conditions**

**Exercice 3.1)**

Écrire un algorithme qui demande un nombre à l’utilisateur, et l’informe ensuite si ce nombre est positif ou négatif (on laisse de côté le cas où le nombre vaut zéro).

**Exercice 3.2)**

Écrire un algorithme qui demande deux nombres à l’utilisateur et l’informe ensuite si leur produit est négatif ou positif (on laisse de côté le cas où le produit est nul). Attention toutefois : on ne doit pas calculer le produit des deux nombres.

**Exercice 3.3)**

Écrire un algorithme qui demande un nombre à l’utilisateur, et l’informe ensuite si ce nombre est positif ou négatif (on inclut cette fois le traitement du cas où le nombre vaut zéro).

**Exercice 3.4)**

Écrire un algorithme qui demande deux nombres à l’utilisateur et l’informe ensuite si le produit est négatif ou positif (on inclut cette fois le traitement du cas où le produit peut être nul). Attention toutefois, on ne doit pas calculer le produit !

**Exercice 3.5)**

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

"Poussin" de 6 à 7 ans

"Pupille" de 8 à 9 ans

"Minime" de 10 à 11 ans

"Cadet" après 12 ans

Peut-on concevoir plusieurs algorithmes équivalents menant à ce résultat ?

**Partie IV) Logique**

**Exercice 4.1)**

Cet algorithme est destiné à prédire l'avenir, et il doit être infaillible !

Il lira au clavier l’heure et les minutes, et il affichera l’heure qu’il sera une minute plus tard. Par exemple, si l'utilisateur tape 21 puis 32, l'algorithme doit répondre :

"Dans une minute, il sera 21 heure(s) 33".

NB : on suppose que l'utilisateur entre une heure valide. Pas besoin donc de la vérifier.

**Exercice 4.2)**

De même que le précédent, cet algorithme doit demander une heure et en afficher une autre. Mais cette fois, il doit gérer également les secondes, et afficher l'heure qu'il sera une seconde plus tard.

Par exemple, si l'utilisateur tape 21, puis 32, puis 8, l'algorithme doit répondre : "Dans une seconde, il sera 21 heure(s), 32 minute(s) et 9 seconde(s)".

NB : là encore, on suppose que l'utilisateur entre une date valide.

**Exercice 4.3)**

Un magasin de reprographie facture 0,10€ les dix premières photocopies, 0,09€ les vingt suivantes et 0,08€ au-delà. Écrivez un algorithme qui demande à l’utilisateur le nombre de photocopies effectuées et qui affiche la facture correspondante.

**Exercice 4.4)**

Les habitants de Zorglub paient l’impôt selon les règles suivantes :

Les hommes de plus de 20 ans paient l’impôt

Les femmes paient l’impôt si elles ont entre 18 et 35 ans

Les autres ne paient pas d’impôt

Le programme demandera donc l’âge et le sexe du Zorglubien, et se prononcera donc ensuite sur le fait que l’habitant est imposable.

**Exercice 4.5)**

Les élections législatives, en Guignolerie Septentrionale, obéissent à la règle suivante :

Lorsque l'un des candidats obtient plus de 50% des suffrages, il est élu dès le premier tour.

En cas de deuxième tour, peuvent participer uniquement les candidats ayant obtenu au moins 12,5% des voix au premier tour.

Vous devez écrire un algorithme qui permette la saisie des scores de quatre candidats au premier tour. Cet algorithme traitera ensuite le candidat numéro 1 (et uniquement lui) : il dira s'il est élu, battu, s'il se trouve en ballottage favorable (il participe au second tour en étant arrivé en tête à l'issue du premier tour) ou défavorable (il participe au second tour sans avoir été en tête au premier tour).

**Exercice 4.6)**

Une compagnie d'assurance automobile propose à ses clients quatre familles de tarifs identifiables par une couleur, du moins au plus onéreux : tarifs bleu, vert, orange et rouge. Le tarif dépend de la situation du conducteur :

Un conducteur de moins de 25 ans et titulaire du permis depuis moins de deux ans, se voit attribuer le tarif rouge, si toutefois il n'a jamais été responsable d'accident. Sinon, la compagnie refuse de l'assurer.

Un conducteur de moins de 25 ans et titulaire du permis depuis plus de deux ans, ou de plus de 25 ans mais titulaire du permis depuis moins de deux ans a le droit au tarif orange s'il n'a jamais provoqué d'accident, au tarif rouge pour un accident, sinon il est refusé.

Un conducteur de plus de 25 ans titulaire du permis depuis plus de deux ans bénéficie du tarif vert s'il n'est à l'origine d'aucun accident et du tarif orange pour un accident, du tarif rouge pour deux accidents, et refusé au-delà

De plus, pour encourager la fidélité des clients acceptés, la compagnie propose un contrat de la couleur immédiatement la plus avantageuse s'il est entré dans la maison depuis plus de cinq ans. Ainsi, s'il satisfait à cette exigence, un client normalement "vert" devient "bleu", un client normalement "orange" devient "vert", et le "rouge" devient orange.

Écrire l'algorithme permettant de saisir les données nécessaires (sans contrôle de saisie) et de traiter ce problème. Avant de se lancer à corps perdu dans cet exercice, on pourra réfléchir un peu et s'apercevoir qu'il est plus simple qu'il n'en a l'air (cela s'appelle faire une analyse !)

**Exercice 4.7)**

Écrivez un algorithme qui après avoir demandé un numéro de jour, de mois et d'année à l'utilisateur, renvoie s'il s'agit ou non d'une date valide.

Cet exercice est certes d’un manque d’originalité affligeant, mais après tout, en algorithmique comme ailleurs, il faut connaître ses classiques ! Et quand on a fait cela une fois dans sa vie, on apprécie pleinement l’existence d’un type numérique « date » dans certains langages…).

Il n'est sans doute pas inutile de rappeler rapidement que le mois de février compte 28 jours, sauf si l’année est bissextile, auquel cas il en compte 29. L’année est bissextile si elle est divisible par quatre. Toutefois, les années divisibles par 100 ne sont pas bissextiles, mais les années divisibles par 400 le sont. Ouf !

Un dernier petit détail : vous ne savez pas, pour l’instant, exprimer correctement l’idée qu’un nombre A est divisible par un nombre B. Aussi, vous vous contenterez d’écrire A%B.

**Partie V) Boucles**

**Exercice 5.1)**

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

**Exercice 5.2)**

Écrire 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.

**Exercice 5.3)**

Écrire 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.

Vous devez utiliser l’instruction while / do while

**Exercice 5.4)**

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

**Exercice 5.5)**

Écrire 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 5.6)**

Écrire 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 5.7)**

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) :

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.

**Partie VI) Tableaux**

**Exercice 6.1)**

Écrire un algorithme qui déclare et remplisse un tableau de 6 valeurs numériques en les mettant toutes à zéro.

**Exercice 6.2)**

Écrire un algorithme qui déclare un tableau de 5 notes, dont on fait ensuite saisir les valeurs par l’utilisateur.

**Exercice 6.3)**

Écrivez un algorithme calculant la somme des valeurs d’un tableau (on suppose que le tableau a été préalablement saisi).

**Exercice 6.4)**

Écrivez un algorithme constituant un tableau, à partir de deux tableaux de même longueur préalablement saisis. Le nouveau tableau sera la somme des éléments des deux tableaux de départ.

Tableau 1 :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 4 | 8 | 7 | 9 | 1 | 5 | 4 | 6 |

Tableau 2 :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 7 | 6 | 5 | 2 | 1 | 3 | 7 | 4 |

Tableau à constituer :

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| 11 | 14 | 12 | 11 | 2 | 8 | 11 | 10 |