

## Avant-propos :

Les exercices et corrigés de ce document sont pris du site Internet : <http://www.pise.info/algo>

# Enoncés des Exercices

## 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    1

B    A + 3

A    3

Fin

---

### Exercice 1.2

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

Variables A, B, C en Entier

Début

A    5

B    3

C    A + B

A    2

C    B – A

Fin

---

### Exercice 1.3

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

Variables A, B en Entier

Début

A    5

B    A + 4

A    A + 1

B    A – 4

Fin

---

### Exercice 1.4

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

Variables A, B, C en Entier

Début

A    3

B    10

C    A + B

B    A + B

A    C

Fin

---

### Exercice 1.5

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

Moralité : les deux dernières instructions permettent-elles d'échanger les deux valeurs de B et A ? Si l'on inverse les deux dernières instructions, cela change-t-il quelque chose ?

---

### Exercice 1.6

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 1.7

Une variante du précédent : on dispose de trois variables A, B et C. Ecrivez 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).

---

### Exercice 1.8

Que produit l'algorithme suivant ?

Variables A, B, C en Caractères

Début

A "423"

B "12"

C A + B

Fin

---

Exercice 1.9

Que produit l'algorithme suivant ?

Variables A, B, C en Caractères

Début

A "423"

B "12"

C A & B

Fin

---

## Lecture et écriture

Exercice 2.1

Quel résultat produit le programme suivant ?

Variables val, double numériques

Début

Val 231

Double Val \* 2

Ecrire Val

Ecrire Double

Fin

---

Exercice 2.2

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

---

Exercice 2.3

Ecrire 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.

---

Exercice 2.4

Ecrire un algorithme utilisant des variables de type chaîne de caractères, et affichant quatre variantes possibles de la célèbre « belle marquise, vos beaux yeux me font mourir d'amour ». On ne se soucie pas de la ponctuation, ni des majuscules.

---

## Les tests

Exercice 3.1

Ecrire 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

Ecrire 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

Ecrire un algorithme qui demande trois noms à l'utilisateur et l'informe ensuite s'ils sont rangés ou non dans l'ordre alphabétique.

---

Exercice 3.4

Ecrire 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 3.5

Ecrire 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 3.6

Ecrire 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 ?

## Les boucles

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.

---

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.

---

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.

---

Exercice 5.4

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 :

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

...

$$7 \times 10 = 70$$

---

Exercice 5.5

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 5.6

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

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

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

---

Exercice 5.7

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 5.8

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.

---

Exercice 5.9

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.

---

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 écrirez une deuxième version permettant de le résoudre.

## Les tableaux

Exercice 6.1

Ecrire un algorithme qui déclare et remplit un tableau de 7 valeurs numériques en les mettant toutes à zéro.

---

Exercice 6.2

Ecrire un algorithme qui déclare et remplit un tableau contenant les six voyelles de l'alphabet latin.

---

Exercice 6.3

Ecrire un algorithme qui déclare un tableau de 9 notes, dont on fait ensuite saisir les valeurs par l'utilisateur.

---

Exercice 6.4

Que produit l'algorithme suivant ?

Tableau Nb(5) en Entier

Variable i en Entier

Début

Pour i 0 à 5

Nb(i) i \* i

i suivant

Pour i 0 à 5

Ecrire Nb(i)

i suivant

Fin

Peut-on simplifier cet algorithme avec le même résultat ?

---

Exercice 6.5

Que produit l'algorithme suivant ?

Tableau N(6) en Entier

Variables i, k en Entier

Début

N(0) 1

Pour k 1 à 6

N(k) N(k-1) + 2

k Suivant

Pour i 0 à 6

Ecrire N(i)

i suivant

Fin

Peut-on simplifier cet algorithme avec le même résultat ?

## Exercice 6.6

Que produit l'algorithme suivant ?

Tableau Suite(7) en Entier

Variable i en Entier

Début

Suite(0) 1

Suite(1) 1

Pour i 2 à 7

    Suite(i) Suite(i-1) + Suite(i-2)

i suivant

Pour i 0 à 7

    Ecrire Suite(i)

i suivant

Fin

## Exercice 6.7

Ecrivez la fin de l'algorithme 6.3 afin que le calcul de la moyenne des notes soit effectué et affiché à l'écran.

## Exercice 6.8

Ecrivez un algorithme permettant à l'utilisateur de saisir un nombre quelconque de valeurs, qui devront être stockées dans un tableau. L'utilisateur doit donc commencer par entrer le nombre de valeurs qu'il compte saisir. Il effectuera ensuite cette saisie. Enfin, une fois la saisie terminée, le programme affichera le nombre de valeurs négatives et le nombre de valeurs positives.

## Exercice 6.9

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

## Exercice 6.10

Ecrivez 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
----	----	----	----	---	---	----	----

## Exercice 6.11

Toujours à partir de deux tableaux précédemment saisis, écrivez un algorithme qui calcule le schtroumpf des deux tableaux. Pour calculer le schtroumpf, il faut multiplier chaque élément du tableau 1 par chaque élément du tableau 2, et additionner le tout. Par exemple si l'on a :

Tableau 1 :

4	8	7	12
---	---	---	----

Tableau 2 :

3	6
---	---

Le Schtroumpf sera :

$$3 * 4 + 3 * 8 + 3 * 7 + 3 * 12 + 6 * 4 + 6 * 8 + 6 * 7 + 6 * 12 = 279$$

## Exercice 6.12

Ecrivez un algorithme qui permette la saisie d'un nombre quelconque de valeurs, sur le principe de l'ex 6.8. Toutes les valeurs doivent être ensuite augmentées de 1, et le nouveau tableau sera affiché à l'écran.

## Exercice 6.13

Ecrivez un algorithme permettant, toujours sur le même principe, à l'utilisateur de saisir un nombre déterminé de valeurs. Le programme, une fois la saisie terminée, renvoie la plus grande valeur en précisant quelle position elle occupe dans le tableau. On prendra soin d'effectuer la saisie dans un premier temps, et la recherche de la plus grande valeur du tableau dans un second temps.

---

Exercice 6.14

Toujours et encore sur le même principe, écrivez un algorithme permettant, à l'utilisateur de saisir les notes d'une classe. Le programme, une fois la saisie terminée, renvoie le nombre de ces notes supérieures à la moyenne de la classe.

## Les fonctions prédéfinies

---

Exercice 9.1

Parmi ces affectations (considérées indépendamment les unes des autres), lesquelles provoqueront des erreurs, et pourquoi ?

Variables A, B, C en Numérique

Variables D, E en Caractère

A Sin(B)

A Sin(A + B \* C)

B Sin(A) – Sin(D)

D Sin(A / B)

C Cos(Sin(A))

---

Exercice 9.2

Ecrivez un algorithme qui demande un mot à l'utilisateur et qui affiche à l'écran le nombre de lettres de ce mot (c'est vraiment tout bête).

---

Exercice 9.3

Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui affiche à l'écran le nombre de mots de cette phrase. On suppose que les mots ne sont séparés que par des espaces (et c'est déjà un petit peu moins bête).

---

Exercice 9.4

Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui affiche à l'écran le nombre de voyelles contenues dans cette phrase.

On pourra écrire deux solutions. La première déploie une condition composée bien fastidieuse. La deuxième, en utilisant la fonction Trouve, allège considérablement l'algorithme.

---

Exercice 9.5

Ecrivez un algorithme qui demande une phrase à l'utilisateur. Celui-ci entrera ensuite le rang d'un caractère à supprimer, et la nouvelle phrase doit être affichée (on doit réellement supprimer le caractère dans la variable qui stocke la phrase, et pas uniquement à l'écran).

---

Exercice 9.6 - Cryptographie 1

Un des plus anciens systèmes de cryptographie (aisément déchiffrable) consiste à décaler les lettres d'un message pour le rendre illisible. Ainsi, les A deviennent des B, les B des C, etc. Ecrivez un algorithme qui demande une phrase à l'utilisateur et qui la code selon ce principe. Comme dans le cas précédent, le codage doit s'effectuer au niveau de la variable stockant la phrase, et pas seulement à l'écran.

---

Exercice 9.7 - Cryptographie 2 - *le chiffre de César*

Une amélioration (relative) du principe précédent consiste à opérer avec un décalage non de 1, mais d'un nombre quelconque de lettres. Ainsi, par exemple, si l'on choisit un décalage de 12, les A deviennent des M, les B des N, etc.

Réalisez un algorithme sur le même principe que le précédent, mais qui demande en plus quel est le décalage à utiliser. Votre sens proverbial de l'élégance vous interdira bien sûr une série de vingt-six "Si...Alors"

---

Exercice 9.8 - Cryptographie 3

Une technique ultérieure de cryptographie consista à opérer non avec un décalage systématique, mais par une substitution aléatoire. Pour cela, on utilise un alphabet-clé, dans lequel les lettres se succèdent de manière désordonnée, par exemple :

HYLUJVPVREAKBNDQFSQZCWMGITX

C'est cette clé qui va servir ensuite à coder le message. Selon notre exemple, les A deviendront des H, les B des Y, les C des L, etc.

Ecrire un algorithme qui effectue ce cryptage (l'alphabet-clé sera saisi par l'utilisateur, et on suppose qu'il effectue une saisie correcte).

---

Exercice 9.9 - Cryptographie 4 - *le chiffre de Vigenère*

Un système de cryptographie beaucoup plus difficile à briser que les précédents fut inventé au XVI<sup>e</sup> siècle par le français Vigenère. Il consistait en une combinaison de différents chiffres de César.

On peut en effet écrire 25 alphabets décalés par rapport à l'alphabet normal :

- l'alphabet qui commence par B et finit par ...YZA



- l'alphabet qui commence par C et finit par ...ZAB
- etc.

Le codage va s'effectuer sur le principe du chiffre de César : on remplace la lettre d'origine par la lettre occupant la même place dans l'alphabet décalé.

Mais à la différence du chiffre de César, un même message va utiliser non un, mais plusieurs alphabets décalés. Pour savoir quels alphabets doivent être utilisés, et dans quel ordre, on utilise une clé.

Si cette clé est "VIGENERE" et le message "Il faut coder cette phrase", on procédera comme suit :

La première lettre du message, I, est la 9e lettre de l'alphabet normal. Elle doit être codée en utilisant l'alphabet commençant par la première lettre de la clé, V. Dans cet alphabet, la 9e lettre est le D. I devient donc D.

La deuxième lettre du message, L, est la 11e lettre de l'alphabet normal. Elle doit être codée en utilisant l'alphabet commençant par la deuxième lettre de la clé, I. Dans cet alphabet, la 11e lettre est le S. L devient donc S, etc.

Quand on arrive à la dernière lettre de la clé, on recommence à la première.

Ecrire l'algorithme qui effectue un cryptage de Vigenère, en demandant bien sûr au départ la clé à l'utilisateur.

---

#### Exercice 9.10

Ecrivez un algorithme qui demande un nombre entier à l'utilisateur. L'ordinateur affiche ensuite le message "Ce nombre est pair" ou "Ce nombre est impair" selon le cas.

---

#### Exercice 9.11

Ecrivez les algorithmes qui génèrent un nombre Glup aléatoire tel que ...

- $0 \leq \text{Glup} < 2$
- $-1 \leq \text{Glup} < 1$
- $1,35 \leq \text{Glup} < 1,65$
- Glup émule un dé à six faces
- $-10,5 \leq \text{Glup} < +6,5$
- Glup émule la somme du jet simultané de deux dés à six faces

## Les fonctions et procédures

#### Exercice 11.1

Écrivez une fonction qui renvoie la somme de cinq nombres fournis en argument.

---

#### Exercice 11.2

Écrivez une fonction qui renvoie le nombre de voyelles contenues dans une chaîne de caractères passée en argument. Au passage, notez qu'une fonction a tout à fait le droit d'appeler une autre fonction.

---

#### Exercice 11.3

Réécrivez la fonction Trouve, vue précédemment, à l'aide des fonctions Mid et Len (comme quoi, Trouve, à la différence de Mid et Len, n'est pas une fonction indispensable dans un langage).

# Corrigés Des Exercices

Les variables

## Exercice 1.1

Après La valeur des variables est :

A	1	A = 1	B = ?
B	A + 3	A = 1	B = 4
A	3	A = 3	B = 4

---

## Exercice 1.2

Après La valeur des variables est :

A	5	A = 5	B = ?	C = ?
B	3	A = 5	B = 3	C = ?
C	A + B	A = 5	B = 3	C = 8
A	2	A = 2	B = 3	C = 8
C	B - A	A = 2	B = 3	C = 1

---

## Exercice 1.3

Après La valeur des variables est :

A	5	A = 5	B = ?
B	A + 4	A = 5	B = 9
A	A + 1	A = 6	B = 9
B	A - 4	A = 6	B = 2

---

## Exercice 1.4

Après La valeur des variables est :

A	3	A = 3	B = ?	C = ?
B	10	A = 3	B = 10	C = ?
C	A + B	A = 3	B = 10	C = 13
B	A + B	A = 3	B = 13	C = 13
A	C	A = 13	B = 13	C = 13

---

## Exercice 1.5

Après La valeur des variables est :

A	5	A = 5	B = ?
B	2	A = 5	B = 2
A	B	A = 2	B = 2
B	A	A = 2	B = 2

Les deux dernières instructions ne permettent donc pas d'échanger les deux valeurs de B et A, puisque l'une des deux valeurs (celle de A) est ici écrasée.

Si l'on inverse les deux dernières instructions, cela ne changera rien du tout, hormis le fait que cette fois c'est la valeur de B qui sera écrasée.

---

## Exercice 1.6

Début

...

C	A
A	B
B	C

Fin

On est obligé de passer par une variable dite temporaire (la variable C).

---

## Exercice 1.7

Début

...

D	C
C	B
B	A
A	D

Fin

En fait, quel que soit le nombre de variables, une seule variable temporaire suffit...

---

## Exercice 1.8

Il ne peut produire qu'une erreur d'exécution, puisqu'on ne peut pas additionner des caractères.

---

## Exercice 1.9

...En revanche, on peut les concaténer. A la fin de l'algorithme, C vaudra donc "42312".

---

## Lecture et écriture

### Exercice 2.1

On verra apparaître à l'écran 231, puis 462 (qui vaut  $231 * 2$ )

---

### Exercice 2.2

Variables nb, carr en Entier

Début

Ecrire "Entrez un nombre :"

Lire nb

carr     $nb * nb$

Ecrire "Son carré est : ", carr

Fin

En fait, on pourrait tout aussi bien économiser la variable carr en remplaçant les deux avant-dernières lignes par :

Ecrire "Son carré est : ",  $nb * nb$

C'est une question de style ; dans un cas, on privilégie la lisibilité de l'algorithme, dans l'autre, on privilégie l'économie d'une variable.

---

### Exercice 2.3

Variables nb, pht, ttva, ptte en Numérique

Début

Ecrire "Entrez le prix hors taxes :"

Lire pht

Ecrire "Entrez le nombre d'articles :"

Lire nb

Ecrire "Entrez le taux de TVA :"

Lire ttva

ptte     $nb * pht * (1 + ttva)$

Ecrire "Le prix toutes taxes est : ", ptte

Fin

Là aussi, on pourrait squeezer une variable et une ligne en écrivant directement. :

Ecrire "Le prix toutes taxes est : ",  $nb * pht * (1 + ttva)$

C'est plus rapide, plus léger en mémoire, mais un peu plus difficile à relire (et à écrire !)

---

### Exercice 2.4

Variables t1, t2, t3, t4 en Caractère

Début

t1    "belle Marquise"

t2    "vos beaux yeux"

t3    "me font mourir"

t4    "d'amour"

Ecrire t1 & " " & t2 & " " & t3 & " " & t4

Ecrire t3 & " " & t2 & " " & t4 & " " & t1

Ecrire t2 & " " & t3 & " " & t1 & " " & t4

Ecrire t4 & " " & t1 & " " & t2 & " " & t3

Fin

## Les tests

### Exercice 3.1

Variable n en Entier

Début

Ecrire "Entrez un nombre : "

Lire n

Si  $n > 0$  Alors

    Ecrire "Ce nombre est positif"

Sinon

    Ecrire "Ce nombre est négatif"

Finsi

Fin

---

### Exercice 3.2

Variables m, n en Entier  
Début  
Ecrire "Entrez deux nombres : "  
Lire m, n  
Si  $(m > 0 \text{ ET } n > 0)$  OU  $(m < 0 \text{ ET } n < 0)$  Alors  
    Ecrire "Leur produit est positif"  
Sinon  
    Ecrire "Leur produit est négatif"  
Finsi  
Fin

---

Exercice 3.3  
Variables a, b, c en Caractère  
Début  
Ecrire "Entrez successivement trois noms : "  
Lire a, b, c  
Si  $a < b$  ET  $b < c$  Alors  
    Ecrire "Ces noms sont classés alphabétiquement"  
Sinon  
    Ecrire "Ces noms ne sont pas classés"  
Finsi  
Fin

---

Exercice 3.4  
Variable n en Entier  
Début  
Ecrire "Entrez un nombre : "  
Lire n  
Si  $n < 0$  Alors  
    Ecrire "Ce nombre est négatif"  
SinonSi  $n = 0$  Alors  
    Ecrire "Ce nombre est nul"  
Sinon  
    Ecrire "Ce nombre est positif"  
Finsi  
Fin

---

Exercice 3.5  
Variables m, n en Entier  
Début  
Ecrire "Entrez deux nombres : "  
Lire m, n  
Si  $m = 0$  OU  $n = 0$  Alors  
    Ecrire "Le produit est nul"  
SinonSi  $(m < 0 \text{ ET } n < 0)$  OU  $(m > 0 \text{ ET } n > 0)$  Alors  
    Ecrire "Le produit est positif"  
Sinon  
    Ecrire "Le produit est négatif"  
Finsi  
Fin

---

Si on souhaite simplifier l'écriture de la condition lourde du SinonSi, on peut toujours passer par des variables booléennes intermédiaires. Une astuce de sioux consiste également à employer un Xor (c'est l'un des rares cas dans lesquels il est pertinent)

---

Exercice 3.6  
Variable age en Entier  
Début  
Ecrire "Entrez l'âge de l'enfant : "  
Lire age  
Si  $\text{age} \geq 12$  Alors  
    Ecrire "Catégorie Cadet"  
SinonSi  $\text{age} \geq 10$  Alors  
    Ecrire "Catégorie Minime"  
SinonSi  $\text{age} \geq 8$  Alors  
    Ecrire "Catégorie Pupille"  
SinonSi  $\text{age} \geq 6$  Alors

Ecrire "Catégorie Poussin"

Finsi

Fin

On peut évidemment écrire cet algorithme de différentes façons, ne serait-ce qu'en commençant par la catégorie la plus jeune.

## Les boucles

Exercice 5.1

Variable N en Entier

Debut

N ← 0

Ecrire "Entrez un nombre entre 1 et 3"

TantQue N < 1 ou N > 3

  Lire N

    Si N < 1 ou N > 3 Alors

      Ecrire "Saisie erronée. Recommencez"

    FinSi

  FinTantQue

Fin

---

Exercice 5.2

Variable N en Entier

Debut

N ← 0

Ecrire "Entrez un nombre entre 10 et 20"

TantQue N < 10 ou N > 20

  Lire N

  Si N < 10 Alors

    Ecrire "Plus grand !"

  SinonSi N > 20 Alors

    Ecrire "Plus petit !"

  FinSi

  FinTantQue

Fin

---

Exercice 5.3

Variables N, i en Entier

Debut

Ecrire "Entrez un nombre : "

  Lire N

Ecrire "Les 10 nombres suivants sont : "

  Pour i ← N + 1 à N + 10

    Ecrire i

  i Suivant

Fin

---

Exercice 5.4

Variables N, i en Entier

Debut

Ecrire "Entrez un nombre : "

  Lire N

Ecrire "La table de multiplication de ce nombre est : "

  Pour i ← 1 à 10

    Ecrire N, " x ", i, " = ", n\*i

  i Suivant

Fin

---

Exercice 5.5

Variables N, i, Som en Entier

Debut

Ecrire "Entrez un nombre : "

  Lire N

  Som ← 0

  Pour i ← 1 à N

```

    Som ← Som + i
i Suivant
Ecrire "La somme est : ", Som
Fin

```

---

Exercice 5.6

Variables N, i, F en Entier

Debut

Ecrire "Entrez un nombre : "

Lire N

F ← 1

Pour i ← 2 à N

    F ← F \* i

i Suivant

Ecrire "La factorielle est : ", F

Fin

---

Exercice 5.7

Variables N, i, PG en Entier

Debut

PG ← 0

Pour i ← 1 à 20

    Ecrire "Entrez un nombre : "

    Lire N

    Si i = 1 ou N > PG Alors

        PG ← N

    FinSi

i Suivant

Ecrire "Le nombre le plus grand était : ", PG

Fin

En ligne 3, on peut mettre n'importe quoi dans PG, il suffit que cette variable soit affectée pour que le premier passage en ligne 7 ne provoque pas d'erreur.

Pour la version améliorée, cela donne :

Variables N, i, PG, IPG en Entier

Debut

PG ← 0

Pour i ← 1 à 20

    Ecrire "Entrez un nombre : "

    Lire N

    Si i = 1 ou N > PG Alors

        PG ← N

        IPG ← i

    FinSi

i Suivant

Ecrire "Le nombre le plus grand était : ", PG

Ecrire "Il a été saisi en position numéro ", IPG

Fin

---

Exercice 5.8

Variables N, i, PG, IPG en Entier

Debut

N ← 1

i ← 0

PG ← 0

TantQue N <> 0

    Ecrire "Entrez un nombre : "

    Lire N

    i ← i + 1

    Si i = 1 ou N > PG Alors

        PG ← N

        IPG ← i

    FinSi

FinTantQue

Ecrire "Le nombre le plus grand était : ", PG

Ecrire "Il a été saisi en position numéro ", IPG  
Fin

---

**Exercice 5.9**

Variables FF, somdue, M, IPG, Reste, Nb10F, Nb5F En Entier

Debut

E 1

somdue 0

TantQue E <> 0

Ecrire "Entrez le montant : "

Lire E

somdue somdue + E

FinTantQue

Ecrire "Vous devez :", E, " euros"

Ecrire "Montant versé :"

Lire M

Reste M - E

Nb10E 0

TantQue Reste >= 10

Nb10E Nb10E + 1

Reste Reste - 10

FinTantQue

Nb5E 0

Si Reste >= 5

Nb5E 1

Reste Reste - 5

FinSi

Ecrire "Rendu de la monnaie :"

Ecrire "Billets de 10 E : ", Nb10E

Ecrire "Billets de 5 E : ", Nb5E

Ecrire "Pièces de 1 E : ", reste

Fin

---

**Exercice 5.10**

Spontanément, on est tenté d'écrire l'algorithme suivant :

Variables N, P, i, Numé, Déno1, Déno2 en Entier

Debut Ecrire "Entrez le nombre de chevaux partants : "

Lire N

Ecrire "Entrez le nombre de chevaux joués : "

Lire P

Numé 1

Pour i 2 à N

Numé Numé \* i

i Suivant

Déno1 1

Pour i 2 à N-P

Déno1 Déno1 \* i

i Suivant

Déno2 1

Pour i 2 à P

Déno2 Déno2 \* i

i Suivant

Ecrire "Dans l'ordre, une chance sur ", Numé / Déno1

Ecrire "Dans le désordre, une sur ", Numé / (Déno1 \* Déno2)

Fin

Cette version, formellement juste, comporte tout de même deux faiblesses.

La première, et la plus grave, concerne la manière dont elle calcule le résultat final. Celui-ci est le quotient d'un nombre par un autre ; or, ces nombres auront rapidement tendance à être très grands. En calculant, comme on le fait ici, d'abord le numérateur, puis ensuite le dénominateur, on prend le risque de demander à la machine de stocker des nombres trop grands pour qu'elle soit capable de les coder (cf. le préambule). C'est d'autant plus bête que rien ne nous oblige à procéder ainsi : on n'est pas obligé de passer par la division de deux très grands nombres pour obtenir le résultat voulu.

La deuxième remarque est qu'on a programmé ici trois boucles successives. Or, en y regardant bien, on peut voir qu'après simplification de la formule, ces trois boucles comportent le même nombre de tours ! (si vous ne me croyez pas, écrivez un exemple de calcul et biffez les nombres identiques au numérateur et au dénominateur). Ce triple calcul (ces trois boucles)



peut donc être ramené(es) à un(e) seul(e). Et voilà le travail, qui est non seulement bien plus court, mais aussi plus performant :

Variables N, P, i, O, F en Entier

Debut

Ecrire "Entrez le nombre de chevaux partants : "

Lire N

Ecrire "Entrez le nombre de chevaux joués : "

Lire P

A 1

B 1

Pour i 1 à P

A A \* (i + N - P)

B B \* i

i Suivant

Ecrire "Dans l'ordre, une chance sur ", A

Ecrire "Dans le désordre, une chance sur ", A / B

Fin

## Les tableaux

Exercice 6.1

Tableau Truc(6) en Numérique

Variable i en Numérique

Debut

Pour i 0 à 6

Truc(i) 0

i Suivant

Fin

---

Exercice 6.2

Tableau Truc(5) en Caractère

Debut

Truc(0) "a"

Truc(1) "e"

Truc(2) "i"

Truc(3) "o"

Truc(4) "u"

Truc(5) "y"

Fin

---

Exercice 6.3

Tableau Notes(8) en Numérique

Variable i en Numérique

Pour i 0 à 8

Ecrire "Entrez la note numéro ", i + 1

Lire Notes(i)

i Suivant

Fin

---

Exercice 6.4

Cet algorithme remplit un tableau avec six valeurs : 0, 1, 4, 9, 16, 25.

Il les écrit ensuite à l'écran. Simplification :

Tableau Nb(5) en Numérique

Variable i en Numérique

Début

Pour i 0 à 5

Nb(i) i \* i

Ecrire Nb(i)

i Suivant

Fin

---

Exercice 6.5

Cet algorithme remplit un tableau avec les sept valeurs : 1, 3, 5, 7, 9, 11, 13.

Il les écrit ensuite à l'écran. Simplification :

Tableau N(6) en Numérique  
 Variables i, k en Numérique  
 Début  
   N(0) ← 1  
   Ecrire N(0)  
   Pour k ← 1 à 6  
     N(k) ← N(k-1) + 2  
     Ecrire N(k)  
   k Suivant  
 Fin

---

Exercice 6.6  
 Cet algorithme remplit un tableau de 8 valeurs : 1, 1, 2, 3, 5, 8, 13, 21

---

Exercice 6.7  
 Variable S en Numérique  
 Tableau Notes(8) en Numérique  
 Debut  
   s ← 0  
   Pour i ← 0 à 8  
     Ecrire "Entrez la note n° ", i + 1  
     Lire Notes(i)  
     s ← s + Notes(i)  
   i Suivant  
   Ecrire "Moyenne :", s/9  
 Fin

---

Exercice 6.8  
 Variables Nb, Nbpos, Nbneg en Numérique  
 Tableau T() en Numérique  
 Debut  
   Ecrire "Entrez le nombre de valeurs :"  
   Lire Nb  
   Redim T(Nb-1)  
   Nbpos ← 0  
   Nbneg ← 0  
   Pour i ← 0 à Nb - 1  
     Ecrire "Entrez le nombre n° ", i + 1  
     Lire T(i)  
     Si T(i) > 0 alors  
       Nbpos ← Nbpos + 1  
     Sinon  
       Nbneg ← Nbneg + 1  
   Finsi  
   i Suivant  
   Ecrire "Nombre de valeurs positives :", Nbpos  
   Ecrire "Nombre de valeurs négatives :", Nbneg  
 Fin

---

Exercice 6.9  
 Variables i, Som, N en Numérique  
 Tableau T() en Numérique  
 Debut  
   ... (on ne programme pas la saisie du tableau, dont on suppose qu'il compte N éléments)  
   Redim T(N-1)  
   ...  
   Som ← 0  
   Pour i ← 0 à N - 1  
     Som ← Som + T(i)  
   i Suivant  
   Ecrire "Somme des éléments du tableau :", Som  
 Fin

---

Exercice 6.10

---

```

Variables i, N en Numérique
Tableaux T1(), T2(), T3() en Numérique
Debut
... (on suppose que T1 et T2 comptent N éléments, et qu'ils sont déjà saisis)
Redim T3(N-1)
...
Pour i    0 à N - 1
    T3(i)  T1(i) + T2(i)
i Suivant
Fin

```

---

Exercice 6.11

```

Variables i, j, N1, N2, S en Numérique
Tableaux T1(), T2() en Numérique
Debut
... On ne programme pas la saisie des tableaux T1 et T2.
On suppose que T1 possède N1 éléments, et que T2 en possède T2)
...
S    0
Pour i    0 à N1 - 1
    Pour j    0 à N2 - 1
        S    S + T1(i) * T2(j)
    j Suivant
i Suivant
Ecrire "Le schtroumpf est : ", S
Fin

```

---

Exercice 6.12

```

Variables Nb, i en Numérique
Tableau T() en Numérique
Debut
Ecrire "Entrez le nombre de valeurs : "
Lire Nb
Redim T(Nb-1)
Pour i    0 à Nb - 1
    Ecrire "Entrez le nombre n° ", i + 1
    Lire T(i)
i Suivant
Ecrire "Nouveau tableau : "
Pour i    0 à Nb - 1
    T(i)  T(i) + 1
    Ecrire T(i)
i Suivant
Fin

```

---

Exercice 6.13

```

Variables Nb, Posmaxi en Numérique
Tableau T() en Numérique
Ecrire "Entrez le nombre de valeurs : "
Lire Nb
Redim T(Nb-1)
Pour i    0 à Nb - 1
    Ecrire "Entrez le nombre n° ", i + 1
    Lire T(i)
i Suivant
Posmaxi  0
Pour i    0 à Nb - 1
    Si T(i) > T(Posmaxi) alors
        Posmaxi  i
    Finsi
i Suivant
Ecrire "Element le plus grand : ", T(Posmaxi)
Ecrire "Position de cet élément : ", Posmaxi
Fin

```

---

Exercice 6.14

```
Variables Nb, i, Som, Moy, Nbsup en Numérique
Tableau T() en Numérique
Debut
Ecrire "Entrez le nombre de notes à saisir : "
Lire Nb
Redim T(Nb-1)
Pour i 0 à Nb - 1
    Ecrire "Entrez le nombre n° ", i + 1
    Lire T(i)
i Suivant
Som 0
Pour i 0 à Nb - 1
    Som Som + T(i)
i Suivant
Moy Som / Nb
NbSup 0
Pour i 0 à Nb - 1
    Si T(i) > Moy Alors
        NbSup NbSup + 1
    FinSi
i Suivant
Ecrire NbSup, " élèves dépassent la moyenne de la classe"
Fin
```

### Les fonctions prédéfinies

#### Exercice 9.1

```
A Sin(B)      Aucun problème
A Sin(A + B * C)  Aucun problème
B Sin(A) – Sin(D) Erreur ! D est en caractère
D Sin(A / B)     Aucun problème... si B est différent de zéro
C Cos(Sin(A))    Erreur ! Il manque une parenthèse fermante
```

---

#### Exercice 9.2

Vous étiez prévenus, c'est bête comme chou ! Il suffit de se servir de la fonction Len, et c'est réglé :

```
Variable Mot en Caractère
Variable Nb en Entier
Debut
Ecrire "Entrez un mot : "
Lire Mot
Nb Len(Mot)
Ecrire "Ce mot compte ", Nb, " lettres"
Fin
```

---

#### Exercice 9.3

Là, on est obligé de compter par une boucle le nombre d'espaces de la phrase, et on en déduit le nombre de mots. La boucle examine les caractères de la phrase un par un, du premier au dernier, et les compare à l'espace.

```
Variable Bla en Caractère
Variables Nb, i en Entier
Debut
Ecrire "Entrez une phrase : "
Lire Bla
Nb 0
Pour i 1 à Len(Bla)
    Si Mid(Bla, i, 1) = " " Alors
        Nb Nb + 1
    FinSi
i suivant
Ecrire "Cette phrase compte ", Nb + 1, " mots"
Fin
```

---

#### Exercice 9.4

Solution 1 : pour chaque caractère du mot, on pose une très douloureuse condition composée. Le moins que l'on puisse dire, c'est que ce choix ne se distingue pas par son élégance. Cela dit, il marche, donc après tout, pourquoi pas.

```

Variable Bla en Caractère
Variables Nb, i, j en Entier
Debut
Ecrire "Entrez une phrase : "
Lire Bla
Nb ← 0
Pour i ← 1 à Len(Bla)
    Si Mid(Bla, i, 1) = "a" ou Mid(Bla, i, 1) = "e" ou Mid(Bla, i, 1) = "i" ou Mid(Bla, i, 1) = "o" ou Mid(Bla, i, 1) = "u" ou
    Mid(Bla, i, 1) = "y" Alors
        Nb ← Nb + 1
    FinSi
i suivant
Ecrire "Cette phrase compte ", Nb, " voyelles"
Fin

```

Solution 2 : on stocke toutes les voyelles dans une chaîne. Grâce à la fonction Trouve, on détecte immédiatement si le caractère examiné est une voyelle ou non. C'est nettement plus sympathique...

```

Variables Bla, Voy en Caractère
Variables Nb, i, j en Entier
Debut
Ecrire "Entrez une phrase : "
Lire Bla
Nb ← 0
Voy ← "aeiouy"
Pour i ← 1 à Len(Bla)
    Si Trouve(Voy, Mid(Bla, i, 1)) > 0 Alors
        Nb ← Nb + 1
    FinSi
i suivant
Ecrire "Cette phrase compte ", Nb, " voyelles"
Fin

```

#### Exercice 9.5

Il n'existe aucun moyen de supprimer directement un caractère d'une chaîne... autrement qu'en procédant par collage. Il faut donc concaténer ce qui se trouve à gauche du caractère à supprimer, avec ce qui se trouve à sa droite. Attention aux paramètres des fonctions Mid, ils n'ont rien d'évident !

```

Variable Bla en Caractère
Variables Nb, i, j en Entier
Début
Ecrire "Entrez une phrase : "
Lire Bla
Ecrire "Entrez le rang du caractère à supprimer : "
Lire Nb
L ← Len(Bla)
Bla ← Mid(Bla, 1, Nb - 1) & Mid(Bla, Nb + 1, L - Nb)
Ecrire "La nouvelle phrase est : ", Bla
Fin

```

#### Exercice 9.6

Sur l'ensemble des exercices de cryptographie, il y a deux grandes stratégies possibles :

- soit transformer les caractères en leurs codes ASCII. L'algorithme revient donc ensuite à traiter des nombres. Une fois ces nombres transformés, il faut les reconvertir en caractères.

- soit en rester au niveau des caractères, et procéder directement aux transformations à ce niveau. C'est cette dernière option qui est choisie ici, et pour tous les exercices de cryptographie à venir.

Pour cet exercice, il y a une règle générale : pour chaque lettre, on détecte sa position dans l'alphabet, et on la remplace par la lettre occupant la position suivante. Seul cas particulier, la vingt-sixième lettre (le Z) doit être codée par la première (le A), et non par la vingt-septième, qui n'existe pas !

```

Variables Bla, Cod, Alpha en Caractère
Variables i, Pos en Entier
Début
Ecrire "Entrez la phrase à coder : "
Lire Bla
Alpha ← "ABCDEFGHIJKLMNOPQRSTUVWXYZ"
Cod ← ""

```

```
Pour i 1 à Len(Bla)
  Let Mid(Bla, i, 1)
  Si Let <> "Z" Alors
    Pos Trouve(Alpha, Let)
    Cod Cod & Mid(Alpha, Pos + 1, 1)
  Sinon
    Cod Cod & "A"
  FinSi
i Suivant
Bla Cod
Ecrire "La phrase codée est : ", Bla
Fin
```

---

## Exercice 9.7

Cet algorithme est une généralisation du précédent. Mais là, comme on ne connaît pas d'avance le décalage à appliquer, on ne sait pas a priori combien de "cas particuliers", à savoir de dépassements au-delà du Z, il va y avoir.

Il faut donc trouver un moyen simple de dire que si on obtient 27, il faut en réalité prendre la lettre numéro 1 de l'alphabet, que si on obtient 28, il faut en réalité prendre la numéro 2, etc. Ce moyen simple existe : il faut considérer le reste de la division par 26, autrement dit le modulo.

Il y a une petite ruse supplémentaire à appliquer, puisque 26 doit rester 26 et ne pas devenir 0.

Variable Bla, Cod, Alpha en Caractère

Variables i, Pos, Décal en Entier

Début

Ecrire "Entrez le décalage à appliquer : "

Lire Décal

Ecrire "Entrez la phrase à coder : "

Lire Bla

Alpha "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

Cod ""

```
Pour i 1 à Len(Bla)
```

```
  Let Mid(Bla, i, 1)
```

```
  Pos Trouve(Alpha, Let)
```

```
  NouvPos Mod(Pos + Décal, 26)
```

```
  Si NouvPos = 0 Alors
```

```
    NouvPos 26
```

```
  FinSi
```

```
  Cod Cod & Mid(Alpha, NouvPos, 1)
```

```
i Suivant
```

```
Bla Cod
```

```
Ecrire "La phrase codée est : ", Bla
```

```
Fin
```

---

## Exercice 9.8

Là, c'est assez direct.

Variable Bla, Cod, Alpha en Caractère

Variables i, Pos, Décal en Entier

Début

Ecrire "Entrez l'alphabet clé : "

Lire Clé

Ecrire "Entrez la phrase à coder : "

Lire Bla

Alpha "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

Cod ""

```
Pour i 1 à Len(Bla)
```

```
  Let Mid(Bla, i, 1)
```

```
  Pos Trouve(Alpha, Let)
```

```
  Cod Cod & Mid(Clé, Pos, 1)
```

```
i Suivant
```

```
Bla Cod
```

```
Ecrire "La phrase codée est : ", Bla
```

```
Fin
```

---

## Exercice 9.9

Le codage de Vigenère n'est pas seulement plus difficile à briser; il est également un peu plus raide à programmer. La difficulté essentielle est de comprendre qu'il faut deux boucles: l'une pour parcourir la phrase à coder, l'autre pour parcourir

la clé. Mais quand on y réfléchit bien, ces deux boucles ne doivent surtout pas être imbriquées. Et en réalité, quelle que soit la manière dont on l'écrit, elle n'en forment qu'une seule.

Variables Alpha, Bla, Cod, Clé, Let en Caractère

Variables i, Pos, PosClé, Décal en Entier

Début

Ecrire "Entrez la clé : "

Lire Clé

Ecrire "Entrez la phrase à coder : "

Lire Bla

Alpha "ABCDEFGHIJKLMNOPQRSTUVWXYZ"

Cod ""

PosClé 0

Pour i 1 à Len(Bla)

On gère la progression dans la clé. J'ai effectué cela "à la main" par une boucle, mais un joli emploi de la fonction Modulo aurait permis une programmation en une seule ligne!

Posclé Posclé + 1

Si PosClé > Len(Clé) Alors

PosClé 1

FinSi

On détermine quelle est la lettre clé et sa position dans l'alphabet

LetClé Mid(Clé, PosClé, 1)

PosLetClé Trouve(Alpha, LetClé)

On détermine la position de la lettre à coder et le décalage à appliquer. Là encore, une solution alternative aurait été d'employer Mod : cela nous aurait épargné le Si...

Let Mid(Bla, i, 1)

Pos Trouve(Alpha, Let)

NouvPos Pos + PosLetClé

Si NouvPos > 26 Alors

NouvPos NouvPos - 26

FinSi

Cod Cod & Mid(Alpha, NouvPos, 1)

i Suivant

Bla Cod

Ecrire "La phrase codée est : ", Bla

Fin

---

#### Exercice 9.10

On en revient à des choses plus simples...

Variable Nb en Entier

Ecrire "Entrez votre nombre : "

Lire Nb

Si Nb/2 = Ent(Nb/2) Alors

Ecrire "Ce nombre est pair"

Sinon

Ecrire "Ce nombre est pair"

FinSi

Fin

---

#### Exercice 9.11

- a) Glup Alea() \* 2
- b) Glup Alea() \* 2 - 1
- c) Glup Alea() \* 0,30 + 1,35
- d) Glup Ent(Alea() \* 6) + 1
- e) Glup Alea() \* 17 - 10,5
- f) Glup Ent(Alea()\*6) + Ent(Alea()\*6) + 2

## Les fonctions et procédures

### Exercice 11.1

Voilà un début en douceur...

Fonction Sum(a, b, c, d, e)

    Renvoyer a + b + c + d + e

FinFonction

---

### Exercice 11.2

Fonction NbVoyelles(Mot en Caractère)

Variables i, nb en Numérique

Pour i    1 à Len(Mot)

    Si Trouve("aeiouy", Mid(Mot, i, 1)) <> 0 Alors

        nb    nb + 1

    FinSi

i suivant

Renvoyer nb

FinFonction

---

### Exercice 11.3

Fonction Trouve(a, b)

Variable i en Numérique

Début

i    1

TantQue i < Len(a) - Len(b) et b <> Mid(a, i, Len(b))

    i    i + 1

FinTantQue

Si b <> Mid(a, i, Len(b)) Alors

    Renvoyer 0

Sinon

    Renvoyer i

FinFonction