

# PARTIE 1

## 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**

# PARTIE 2

## 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.

## PARTIE 3

### 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 ?

## PARTIE 4

### Exercice 4.1

Formulez un algorithme équivalent à l'algorithme suivant :

```
Si Tutu > Toto + 4 OU Tata = "OK" Alors
    Tutu ← Tutu + 1
Sinon
    Tutu ← Tutu - 1
Finsi
```

### Exercice 4.2

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 4.3

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 4.4

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

### Exercice 4.5

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 4.6

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 4.7

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 d'un an.

Ecrire 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 4.8

Ecrivez un algorithme qui a près 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 en pseudo-code l'idée qu'un nombre A est divisible par un nombre B. Aussi, vous vous contenterez d'écrire en bons télégraphistes que A divisible par B se dit « A dp B ».

## PARTIE 5

#### 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 x 1 = 7  
7 x 2 = 14  
7 x 3 = 21  
...  
7 x 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.

## PARTIE 6

### 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 **suisant**

**Pour** i ← 0 à 5

**Ecrire** Nb(i)

i **suisant**

**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 **Suisant**

**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.

# PARTIE 7

## Exercice 7.1

Ecrivez un algorithme qui permette de saisir un nombre quelconque de valeurs, et qui les range au fur et à mesure dans un tableau. Le programme, une fois la saisie terminée, doit dire si les éléments du tableau sont tous consécutifs ou non.

Par exemple, si le tableau est :

12	13	14	15	16	17	18
----	----	----	----	----	----	----

ses éléments sont tous consécutifs. En revanche, si le tableau est :

9	10	11	15	16	17	18
---	----	----	----	----	----	----

ses éléments ne sont pas tous consécutifs.

## Exercice 7.2

Ecrivez un algorithme qui trie un tableau dans l'ordre décroissant.

Vous écrivez bien entendu deux versions de cet algorithme, l'une employant le tri par insertion, l'autre le tri à bulles.

## Exercice 7.3

Ecrivez un algorithme qui inverse l'ordre des éléments d'un tableau dont on suppose qu'il a été préalablement saisi (« les premiers seront les derniers... »)

## Exercice 7.4

Ecrivez un algorithme qui permette à l'utilisateur de supprimer une valeur d'un tableau préalablement saisi. L'utilisateur donnera l'indice de la valeur qu'il souhaite supprimer. Attention, il ne s'agit pas de remettre une valeur à zéro, mais bel et bien de la supprimer du tableau lui-même ! Si le tableau de départ était :

12	8	4	45	64	9	2
----	---	---	----	----	---	---

Et que l'utilisateur souhaite supprimer la valeur d'indice 4, le nouveau tableau sera :

12	8	4	45	9	2
----	---	---	----	---	---

## Exercice 7.5

Ecrivez l'algorithme qui recherche un mot saisi au clavier dans un dictionnaire. Le dictionnaire est supposé être codé dans un tableau préalablement rempli et trié.

# PARTIE 8

## Exercice 8.1

Écrivez un algorithme remplissant un tableau de 6 sur 13, avec des zéros.

## Exercice 8.2

Quel résultat produira cet algorithme ?

**Tableau** x(1, 2) **en Entier**  
**Variables** i, j, val **en Entier**

**Début**  
val ← 1  
**Pour** i ← 0 à 1  
    **Pour** j ← 0 à 2  
        x(i, j) ← val  
        val ← val + 1  
    j **Suivant**  
i **Suivant**  
**Pour** i ← 0 à 1  
    **Pour** j ← 0 à 2  
        **Ecrire** x(i, j)  
    j **Suivant**  
i **Suivant**  
**Fin**

## Exercice 8.3

Quel résultat produira cet algorithme ?

**Tableau** x(1, 2) **en Entier**  
**Variables** i, j, val **en Entier**

**Début**  
val ← 1  
**Pour** i ← 0 à 1  
    **Pour** j ← 0 à 2  
        x(i, j) ← val  
        val ← val + 1  
    j **Suivant**  
i **Suivant**  
**Pour** j ← 0 à 2  
    **Pour** i ← 0 à 1  
        **Ecrire** x(i, j)  
    i **Suivant**  
j **Suivant**  
**Fin**

## Exercice 8.4

Quel résultat produira cet algorithme ?

**Tableau** T(3, 1) **en Entier**  
**Variables** k, m, **en Entier**

**Début**  
**Pour** k ← 0 à 3

```

Pour m ← 0 à 1
  T(k, m) ← k + m
  m Suivant
k Suivant
Pour k ← 0 à 3
  Pour m ← 0 à 1
    Ecrire T(k, m)
  m Suivant
k Suivant
Fin

```

### Exercice 8.5

Mêmes questions, en remplaçant la ligne :

```

T(k, m) ← k + m

par

T(k, m) ← 2 * k + (m + 1)

puis par :

T(k, m) ← (k + 1) + 4 * m

```

### Exercice 8.6

Soit un tableau T à deux dimensions (12, 8) préalablement rempli de valeurs numériques.

Écrire un algorithme qui recherche la plus grande valeur au sein de ce tableau.

### Exercice 8.7

Écrire un algorithme de jeu de dames très simplifié.

L'ordinateur demande à l'utilisateur dans quelle case se trouve son pion (quelle ligne, quelle colonne). On met en place un contrôle de saisie afin de vérifier la validité des valeurs entrées.

Ensuite, on demande à l'utilisateur quel mouvement il veut effectuer : 0 (en haut à gauche), 1 (en haut à droite), 2 (en bas à gauche), 3 (en bas à droite).

Si le mouvement est impossible (i.e. on sort du damier), on le signale à l'utilisateur et on s'arrête là. Sinon, on déplace le pion et on affiche le damier résultant, en affichant un « O » pour une case vide et un « X » pour la case où se trouve le pion.

## PARTIE 9

### 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 :

HYLUJVPREAKBNDOFSQZCWMGITX

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 XVIe 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 12e 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 12e 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

## PARTIE 10

### Exercice 10.1

Quel résultat cet algorithme produit-il ?

**variable** Truc en Caractère

**Début**

**Ouvrir** "Exemple.txt" sur 5 en Lecture

**Tantque** Non EOF(5)

**LireFichier** 5, Truc

**Ecrire** Truc

**FinTantQue**

**Fermer** 5

**Fin**

### Exercice 10.2

Ecrivez l'algorithme qui produit un résultat similaire au précédent, mais le fichier texte "Exemple.txt" est cette fois de type délimité (caractère de délimitation : /). On produira à l'écran un affichage où pour des raisons esthétiques, ce caractère sera remplacé avec des espaces.

### Exercice 10.3

On travaille avec le fichier du carnet d'adresses en champs de largeur fixe.

Ecrivez un algorithme qui permet à l'utilisateur de saisir au clavier un nouvel individu qui sera ajouté à ce carnet d'adresses.

### Exercice 10.4

Même question, mais cette fois le carnet est supposé être trié par ordre alphabétique. L'individu doit donc être inséré au bon endroit dans le fichier.



### Exercice 10.5

Ecrivez un algorithme qui permette de modifier un renseignement (pour simplifier, disons uniquement le nom de famille) d'un membre du carnet d'adresses. Il faut donc demander à l'utilisateur quel est le nom à modifier, puis quel est le nouveau nom, et mettre à jour le fichier. Si le nom recherché n'existe pas, le programme devra le signaler.

### Exercice 10.6

Ecrivez un algorithme qui trie les individus du carnet d'adresses par ordre alphabétique.

### Exercice 10.7

Soient Toto.txt et Tata.txt deux fichiers dont les enregistrements ont la même structure. Ecrire un algorithme qui recopie tout le fichier Toto dans le fichier Tutu, puis à sa suite, tout le fichier Tata (concaténation de fichiers).

### Exercice 10.8

Ecrire un algorithme qui supprime dans notre carnet d'adresses tous les individus dont le mail est invalide (pour employer un critère simple, on considèrera que sont invalides les mails ne comportant aucune arobase, ou plus d'une arobase).

### Exercice 10.9

Les enregistrements d'un fichier contiennent les deux champs Nom (chaîne de caractères) et Montant (Entier). Chaque enregistrement correspond à une vente conclue par un commercial d'une société.

On veut mémoriser dans un tableau, puis afficher à l'écran, le total de ventes par vendeur. Pour simplifier, on suppose que le fichier de départ est déjà trié alphabétiquement par vendeur.

## PARTIE 11

### 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).