ALGORITHMIQUE
&
PROGRAMMATION
PAR LA PRATIQUE

Travaux pratiques résolus
et programmation avec les logiciels Scilab et Python

José OUIN
Ingénieur INSA Toulouse
Professeur agrégé de Génie civil
Professeur agrégé de Mathématiques

ellipses

Avant-Propos

Cet ouvrage propose un ensemble de travaux pratiques portant sur des exemples concrets qui permettent de passer directement à la programmation.

Pour chaque travail pratique, la méthode numérique est rappelée et la liste des fonctions nécessaires est fournie. Toutes les fonctions utilisées sont détaillées en fin d'ouvrage avec des exemples d'utilisation.

Les programmes sont écrits en langages Scilab et Python. La partie relative à la présentation de ces logiciels indique les versions utilisées, les adresses de téléchargement et les modules nécessaires au fonctionnement des programmes.

J'espère que cet ouvrage aidera les enseignants dans leurs missions d'enseignement, les élèves dans leur découverte de l'algorithmique, les étudiants dans leur apprentissage des outils de calcul scientifique ainsi que tous ceux qui souhaitent découvrir l'algorithmique ou approfondir leurs connaissances dans ce domaine.

Je dédie ce livre à tous les passionnés, à tous ceux qui le deviendront.

José OUIN.

Le site Internet http://www.joseouin.fr propose les éléments suivants :

- un didacticiel sur l'installation des modules pour Python ;
- les codes sources Scilab et Python des programmes de cet ouvrage ;
- des ressources complémentaires sur Scilab et Python.

www.joseouin.fr

Sommaire

Chapitre 1. L'algorithmique

Chapitre 2. Le logiciel Scilab

Chapitre 3. Le logiciel Python

Chapitre 4. Les travaux pratiques

Chapitre 5. Les instructions et fonctions du langage Scilab

Chapitre 6. Les instructions et fonctions du langage Python

Chapitre 1 : L'algorithmique

1- Introduction à l'algorithmique

1-1. Définition de l'algorithmique

L'algorithmique désigne l'ensemble des règles et des techniques qui sont impliquées dans la définition et la conception des algorithmes.

Définition d'un algorithme :
Un algorithme est suite finie d'opérations élémentaires constituant un schéma de calcul ou de résolution d'un problème donné.
Remarque :
Le résultat doit s'obtenir en un temps fini. Les opérations élémentaires correspondent à une suite "d'instructions" visant à transformer les données afin d'obtenir le résultat visé.

1-2. Algorithmique et programmation

1-2.1 Un algorithme puis un programme

Un algorithme n'est pas un programme. Un algorithme résout un problème donné indépendamment des particularités du langage de programmation.
Un programme est la réalisation (l'implémentation) d'un algorithme au moyen d'un langage donné (Visual Basic, C++, Scilab, Python, Fortran, etc.). Il s'agit de la mise en œuvre de l'ensemble des instructions.

Par exemple, lors de la programmation il faut définir le format des variables utilisées, ce qui est un problème d'implémentation ignoré au niveau de l'algorithme.

Le "code source" désigne le texte, en langage de programmation, constituant le programme.

1-2.2 Les éléments de base d'un algorithme

- **La préparation du traitement**

Il s'agit de repérer les données nécessaires à la résolution du problème.
Ces données sont de plusieurs types :
Données numériques :

- données saisies au clavier par l'utilisateur ;
- lecture d'un fichier contenant des nombres.

Données graphiques :

- lecture de la position du pointeur de la souris.

Données sous forme de textes :

- données saisies au clavier par l'utilisateur ;
- lecture d’un fichier contenant des nombres.

Il s’agit également de repérer les résultats intermédiaires qu’il est bon de mémoriser pour la suite car indispensables au traitement. Il est parfois utile d’utiliser des variables auxiliaires afin de ne pas modifier les données initiales.

- **Le traitement**

Il s’agit de déterminer toutes les "instructions" à donner pour automatiser la résolution du problème et obtenir les résultats attendus.

- **La sortie des résultats**

Les résultats obtenus peuvent être :

- affichés à l’écran (nombres, textes, graphiques) ;
- imprimés sur papier ;
- écrits dans un fichier.

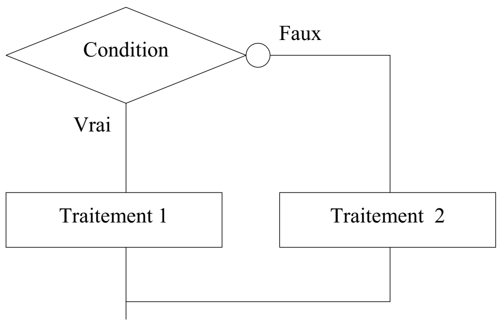
1-2.3 Les conventions d'écriture d'un algorithme

- **L'algorithme graphique ou organigramme de programmation**

Il s'agit d'une représentation graphique comportant des symboles de traitement et de test (rectangles, losanges, etc.) reliés entre eux par des lignes de liaison indiquant le flux des contrôles.
Une norme ISO a été développée, elle porte le numéro ISO 5807. Elle décrit en détails les différents symboles à utiliser pour représenter un algorithme informatique de manière normalisée.

Exemple :
L'organigramme ci-contre représente une structure alternative :

Si la "Condition" est vérifiée **Alors**
Effectuer le "Traitement 1".
Sinon
Effectuer le "Traitement 2".
FinSi



- **L'algorithme textuel ou pseudo-code**

Le pseudo-code est une façon de décrire un algorithme sans référence à un langage de programmation en particulier. Il ressemble cependant à un langage de programmation authentique mais dont on aurait retiré la plupart des problèmes de syntaxe.

Chapitre 4. Les travaux pratiques

Ensemble des diviseurs positifs d'un entier naturel

1- Énoncé

Un diviseur positif d'un entier naturel *n* est un entier naturel *d* tel que la division euclidienne de *n* par *d* donne un reste nul.

• **Exemple**
7 est un diviseur de 42 parce que 42 = 7x6 + 0. On dit que 7 divise 42 ou que 42 est divisible par 7. On dit également que 42 est un multiple de 7.
Les diviseurs peuvent être positifs ou négatifs.
Les diviseurs positifs de 42 sont : {1, 2, 3, 6, 7, 14, 21, 42}.

2- Travail demandé

1) Ecrire l'algorithme permettant de déterminer l'ensemble des diviseurs d'un entier naturel *n* donné.
2) Ecrire le programme correspondant permettant de saisir la valeur de *n*.
3) Déterminer la liste des diviseurs des entiers suivants 287527 et 29312156.

3- Les fonctions à utiliser

3-1. Fonctions Scilab

disp() ; input() ; modulo() ; printf() ; function()

3-2. Fonctions Python

print() ; input() ; a%b ; range() ; eval() ; def()

4- Solution

4-1. L'algorithme

Entrées

```
Saisir la valeur de n
```

Etude de nombres rationnels

1- Énoncé

Un nombre rationnel est un nombre qui peut s'exprimer comme le quotient de deux entiers relatifs :

$$r = \frac{p}{q} \;;\; p \in \mathbb{Z} \text{ et } q \in \mathbb{Z}$$

Lorsque p et q n'ont pas de diviseurs communs autre que 1, on dit que la fraction est irréductible. On dit également que ces nombres sont premiers entre eux car ils vérifient l'égalité :

$$\text{PGCD}(p\,;q) = 1$$

où PGCD désigne le Plus Grand Commun Diviseur des entiers p et q.

Le développement décimal d'un nombre rationnel est toujours périodique au bout d'une certaine décimale. Réciproquement, si un nombre possède un développement décimal périodique, alors c'est un nombre rationnel.

• **Exemple**

$r = 1{,}12345123451234512345123451234512345...$

On écrit les égalités suivantes :

$$(r-1)\times 10^5 - 12345 = r - 1$$

$$(r-1) = \frac{12345}{10^5 - 1}$$

$$r = 1 + \frac{12345}{10^5 - 1}$$

$$r = \frac{10^5 - 1 + 12345}{10^5 - 1}$$

$$r = \frac{112344}{99999}$$

Cette fraction n'est pas irréductible car $\text{PGCD}(112344\,;99999)=3$. En divisant le numérateur et le dénominateur par 3, on obtient :

$$r=\frac{37448}{33333}$$

2- Travail demandé

1) Soit r un nombre rationnel positif donné, écrire l'algorithme permettant de déterminer les deux entiers p et q tels que :

$$\begin{cases} r=\dfrac{p}{q} \\ \text{PGCD}(p\,;q)=1 \end{cases}$$

Il s'agit de créer un programme renvoyant un couple de nombres premiers entre eux dont le quotient est égal à r. L'utilisateur doit indiquer les deux paramètres suivants :
– la troncature à l'unité de r ;
– la séquence périodique des décimales.

Exemple : $r=1{,}12345123451234512345123451234512345...$

Les paramètres sont dans ce cas : 1 et 12345.

2) Ecrire le programme correspondant.
3) A l'aide de ce programme, déterminer les entiers p et q dans les cas suivants :
3.1) $r=1{,}12345123451234512345123451234512345...$
3.2) $r=15{,}2589258925892589...$
3.3) $r=0{,}369369369369...$

3- Les fonctions à utiliser

3-1. Fonctions Scilab

input() ; length() ; printf() ; modulo()

3-2. Fonctions Python

print() ; input() ; a%b ; range() ; eval() ; len()

TP 10 — Formule du binôme de Newton et triangle de Pascal

1- Énoncé

La formule du binôme de Newton est une formule mathématique donnant le développement d'une puissance entière quelconque d'un binôme.

$$(a+b)^n = \sum_{k=0}^{n} \binom{n}{k} \cdot a^{n-k} \cdot b^k \quad \text{, avec } \binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

• Exemple

$$(a+b)^3 = \binom{3}{0} a^3 + \binom{3}{1} a^2.b + \binom{3}{2} a.b^2 + \binom{3}{3} b^3$$

$$(a+b)^3 = a^3 + 3a^2.b + 3a.b^2 + b^3$$

Les coefficients $\binom{n}{k}$ sont donnés dans le triangle de Pascal suivant :

k	0	1	2	3	4	5	6	7	8	9
$n=1$	1	1								
$n=2$	1	2	1							
$n=3$	1	3	3	1						
$n=4$	1	4	6	4	1					
$n=5$	1	5	10	10	5	1				
$n=6$	1	6	15	20	15	6	1			
$n=7$	1	7	21	35	35	21	7	1		
$n=8$	1	8	28	56	70	56	28	8	1	
$n=9$	1	9	36	84	126	126	84	36	9	1

• **Remarque**
On obtient les coefficients d'une ligne donnée en effectuant les sommes de deux coefficients de la ligne précédente comme indiqué dans le tableau ci-avant.
Par exemple : $\binom{5}{2} = \binom{4}{2} + \binom{4}{1} = 6 + 4 = 10$; $\binom{7}{4} = \binom{6}{4} + \binom{6}{3} = 15 + 20 = 35$.

Les cases grisées du tableau ci-dessous indiquent les sommes des coefficients du triangle de Pascal suivant les diagonales. Par exemple :
– pour $n = 6$: 13 = 1 + 5 + 6 + 1 ;
– pour $n = 7$: 21 = 1 + 6 + 10 + 4 ;
– pour $n = 8$: 34 = 1 + 7 + 15 + 10 + 1.

k	0	1	2	3	4	5	6	7	8	9
$n=1$	1	1	**3**	**5**						
$n=2$	1	2	1	**8**	**13**					
$n=3$	1	3	3	1	**21**	**34**				
$n=4$	1	4	6	4	1	**55**	**89**			
$n=5$	1	5	10	10	5	1	**144**	**233**		
$n=6$	1	6	15	20	15	6	1	**377**	**610**	
$n=7$	1	7	21	35	35	21	7	1	**987**	**1597**
$n=8$	1	8	28	56	70	56	28	8	1	**2584**
$n=9$	1	9	36	84	126	126	84	36	9	1
…	…	…	…	…	…	…	…	…	…	…

Il s'agit de calculer les valeurs de ces sommes de termes pour différentes valeurs de *n*.

2- Travail demandé

Pour une valeur de n donnée, on note $S(n)$ la somme des termes de la diagonale. Par exemple, $S(9) = 55$. On donne l'expression de $S(n)$ suivant la parité de n :

Cas où n est pair :

$$S(n) = \sum_{k=0}^{k=\frac{n}{2}} \binom{n-k}{k}$$

TP 20

Ecriture d'un entier naturel dans le système hexadécimal

1- Énoncé

Ce travail pratique propose de déterminer un algorithme permettant d'effectuer un changement de système de numération. Il s'agit de passer de l'écriture d'un entier naturel dans système décimal à l'écriture de ce même nombre dans le système hexadécimal c'est-à-dire en base 16.
Par exemple, le nombre $a = 705$ en base 10 s'écrit $a = \overline{2\mathrm{C}1}_{16}$ en base 16.

Le système hexadécimal nécessite l'introduction de 10 chiffres et de 6 lettres, représentant les 16 premiers entiers naturels de ce système de numération :

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F

• Définition
On rappelle la définition de l'écriture dans une base donnée :
On considère la nombre $a = \overline{a_n a_{n-1 \ldots\ldots} a_3 a_2 a_1 a_0}_{\,b}$ exprimé dans une base b donnée. La valeur de a en base 10 est égale à la somme des termes suivants :

$$a = a_0 + b(a_1) + (b)^2(a_2) + (b)^3(a_3) + \ldots + (b)^{n-1}(a_{n-1}) + (b)^n(a_n)$$

Pour $a = \overline{2C1}_{16}$ en base 16 :

$$a = 1 + 16(12) + (16)^2(2)$$

$$a = 705$$

Remarque : le coefficient C a été remplacé par 12 pour le calcul de la valeur de a en base 10.

• Calcul des différents coefficients
On rappelle l'égalité ci-avant :

$$a = a_0 + b(a_1) + (b)^2(a_2) + (b)^3(a_3) + \ldots + (b)^{n-1}(a_{n-1}) + (b)^n(a_n)$$

TP 25

Détermination des décimales du nombre Pi

1- Énoncé

Le nombre Pi, noté π, est le rapport constant de la circonférence d'un cercle à son diamètre. Ce travail pratique consiste à déterminer les n premières décimales de Pi en base 10. On utilise pour cela la formule d'Euler en écrivant le développement en série de la fonction $\arctan$ vérifiant l'égalité suivante :

$$\frac{\pi}{4} = \arctan(1)$$

On obtient :

$$\pi = 2\sum_{k=0}^{k=\infty} \frac{2^k (k!)^2}{(2k+1)!} = 2 + 2\sum_{k=1}^{k=\infty} \frac{1\times 2\times 3\times 4\times 5\times 6\times 7\times 8\times \ldots\times k}{3\times 5\times 7\times 9\times 11\times \ldots\times (2k+1)}$$

On écrit les premiers termes de cette égalité :

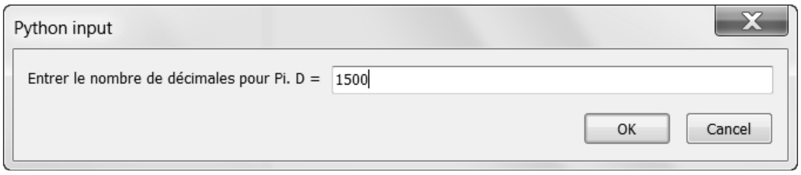
$$\pi = 2 + 2\frac{1}{3} + 2\frac{2}{3\times 5} + 2\frac{2\times 3}{3\times 5\times 7} + 2\frac{2\times 3\times 4}{3\times 5\times 7\times 9} + \ldots + 2\frac{2\times 3\times 4\times 5\times \ldots\times k}{3\times 5\times 7\times 9\ldots\times (2k+1)}$$

L'algorithme de Ruffini-Horner permet d'écrire la forme factorisée suivante :

$$\pi = 1\left(2 + \frac{1}{3}\left(2 + \frac{2}{5}\left(2 + \frac{3}{7}\left(2 + \frac{4}{9}\left(2 + \frac{5}{11}\left(2 + \ldots + \frac{k}{2k+1}(2)\right)\right)\right)\right)\right)\right)$$

On remarque que le nombre π s'écrit $(2;2;2;2;2;2;\ldots\ldots;2)$ dans la base à pas variable $\left[1;\frac{1}{3};\frac{2}{5};\frac{3}{7};\frac{4}{9};\frac{5}{11};\ldots;\frac{k}{2k+1}\right]$.

Il s'agit donc de déterminer les décimales de Pi en base 10 à partir de l'écriture de ce nombre dans une base à pas variable.



Décimales de Pi : [3, 1, 4, 1, 5, 9, 2, 6, 5, 3, 5, 8, 9, 7, 9, 3, 2, 3, 8, 4, 6, 2, 6, 4, 3, 3, 8, 3, 2, 7, 9, 5, 0, 2, 8, 8, 4, 1, 9, 7, 1, 6, 9, 3, 9, 9, 3, 7, 5, 1, 0, 5, 8, 2, 0, 9, 7, 4, 9, 4, 4, 5, 9, 2, 3, 0, 7, 8, 1, 6, 4, 0, 6, 2, 8, 6, 2, 0, 8, 9, 9, 8, 6, 2, 8, 0, 3, 4, 8, 2, 5, 3, 4, 2, 1, 1, 7, 0, 6, 7, 9, 8, 2, 1, 4, 8, 0, 8, 6, 5, 1, 3, 2, 8, 2, 3, 0, 6, 6, 4, 7, 0, 9, 3, 8, 4, 4, 6, 0, 9, 5, 5, 0, 5, 8, 2, 2, 3, 1, 7, 2, 5, 3, 5, 9, 4, 0, 8, 1, 2, 8, 4, 8, 1, 1, 1, 7, 4, 5, 0, 2, 8, 4, 1, 0, 2, 7, 0, 1, 9, 3, 8, 5, 2, 1, 1, 0, 5, 5, 5, 9, 6, 4, 4, 6, 2, 2, 9, 4, 8, 9, 5, 4, 9, 3, 0, 3, 8, 1, 9, 6, 4, 4, 2, 8, 8, 1, 0, 9, 7, 5, 6, 6, 5, 9, 3, 3, 4, 4, 6, 1, 2, 8, 4, 7, 5, 6, 4, 8, 2, 3, 3, 7, 8, 6, 7, 8, 3, 1, 6, 5, 2, 7, 1, 2, 0, 1, 9, 0, 9, 1, 4, 5, 6, 4, 8, 5, 6, 6, 9, 2, 3, 4, 6, 0, 3, 4, 8, 6, 1, 0, 4, 5, 4, 3, 2, 6, 6, 4, 8, 2, 1, 3, 3, 9, 3, 6, 0, 7, 2, 6, 0, 2, 4, 9, 1, 4, 1, 2, 7, 3, 7, 2, 4, 5, 8, 7, 0, 0, 6, 6, 0, 6, 3, 1, 5, 5, 8, 8, 1, 7, 4, 8, 8, 1, 5, 2, 0, 9, 2, 0, 9, 6, 2, 8, 2, 9, 2, 5, 4, 0, 9, 1, 7, 1, 5, 3, 6, 4, 3, 6, 7, 8, 9, 2, 5, 9, 0, 3, 5, 10, 0, 1, 1, 3, 3, 0, 5, 3, 0, 5, 4, 8, 8, 2, 0, 4, 6, 6, 5, 2, 1, 3, 8, 4, 1, 4, 6, 9, 5, 1, 9, 4, 1, 5, 1, 1, 6, 0, 9, 4, 3, 3, 0, 5, 7, 2, 7, 0, 3, 6, 5, 7, 5, 9, 5, 9, 1, 9, 5, 3, 0, 9, 2, 1, 8, 6, 1, 1, 7, 3, 8, 1, 9, 3, 2, 6, 1, 1, 7, 9, 3, 1, 0, 5, 1, 1, 8, 5, 4, 8, 0, 7, 4, 4, 6, 2, 3, 7, 9, 9, 6, 2, 7, 4, 9, 5, 6, 7, 3, 5, 1, 8, 8, 5, 7, 5, 2, 7, 2, 4, 8, 9, 1, 2, 2, 7, 9, 3, 8, 1, 8, 3, 0, 1, 1, 9, 4, 9, 1, 2, 9, 8, 3, 3, 6, 7, 3, 3, 6, 2, 4, 4, 0, 6, 5, 6, 6, 4, 3, 0, 8, 6, 0, 2, 1, 3, 9, 4, 9, 4, 6, 3, 9, 5, 2, 2, 4, 7, 3, 7, 1, 9, 0, 7, 0, 2, 1, 7, 9, 8, 6, 0, 9, 4, 3, 7, 0, 2, 7, 7, 0, 5, 3, 9, 2, 1, 7, 1, 7, 6, 2, 9, 3, 1, 7, 6, 7, 5, 2, 3, 8, 4, 6, 7, 4, 8, 1, 8, 4, 6, 7, 6, 6, 9, 4, 0, 5, 1, 3, 1, 10, 0, 0, 5, 6, 8, 1, 2, 7, 1, 4, 5, 2, 6, 3, 5, 6, 0, 8, 2, 7, 7, 8, 5, 7, 7, 1, 3, 4, 2, 7, 5, 7, 7, 8, 9, 6, 0, 9, 1, 7, 3, 6, 3, 7, 1, 7, 8, 7, 2, 1, 4, 6, 8, 4, 4, 0, 9, 0, 1, 2, 2, 4, 9, 5, 3, 4, 3, 0, 1, 4, 6, 5, 4, 9, 5, 8, 5, 3, 7, 1, 0, 5, 0, 7, 9, 2, 2, 7, 9, 6, 8, 9, 2, 5, 8, 9, 2, 3, 5, 4, 2, 0, 1, 9, 9, 5, 6, 1, 1, 2, 1, 2, 9, 0, 2, 1, 9, 6, 0, 8, 6, 4, 0, 3, 4, 4, 1, 8, 1, 5, 9, 8, 1, 3, 6, 2, 9, 7, 7, 4, 7, 7, 1, 3, 0, 9, 9, 6, 0, 5, 1, 8, 7, 0, 7, 2, 1, 1, 3, 4, 9, 9, 9, 9, 9, 9, 8, 3, 7, 2, 9, 7, 8, 0, 4, 9, 9, 5, 1, 0, 5, 9, 7, 3, 1, 7, 3, 2, 8, 1, 6, 0, 9, 6, 3, 1, 8, 5, 9, 5, 0, 2, 4, 4, 5, 9, 4, 5, 5, 3, 4, 6, 9, 0, 8, 3, 0, 2, 6, 4, 2, 5, 2, 2, 3, 0, 8, 2, 5, 3, 3, 4, 4, 6, 8, 5, 0, 3, 5, 2, 6, 1, 9, 3, 1, 1, 8, 8, 1, 7, 1, 0, 0, 9, 10, 0, 3, 1, 3, 7, 8, 3, 8, 7, 5, 2, 8, 8, 6, 5, 8, 7, 5, 3, 3, 2, 0, 8, 3, 8, 1, 4, 2, 0, 6, 1, 7, 1, 7, 7, 6, 6, 9, 1, 4, 7, 3, 0, 3, 5, 9, 8, 2, 5, 3, 4, 9, 0, 4, 2, 8, 7, 5, 5, 4, 6, 8, 7, 3, 1, 1, 5, 9, 5, 6, 2, 8, 6, 3, 8, 8, 2, 3, 5, 3, 7, 8, 7, 5, 9, 3, 7, 5, 1, 9, 5, 7, 7, 8, 1, 8, 5, 7, 7, 8, 0, 5, 3, 2, 1, 7, 1, 2, 2, 6, 8, 0, 6, 6, 1, 2, 10, 0, 1, 9, 2, 7, 8, 7, 6, 6, 1, 1, 1, 9, 5, 9, 0, 9, 2, 1, 6, 4, 2, 0, 1, 9, 8, 9, 3, 8, 0, 9, 5, 2, 5, 7, 2, 0, 1, 0, 6, 5, 4, 8, 5, 8, 6, 3, 2, 7, 8, 8, 6, 5, 9, 3, 6, 1, 5, 3, 3, 8, 1, 8, 2, 7, 9, 6, 8, 2, 3, 0, 3, 0, 1, 9, 5, 2, 0, 3, 5, 3, 0, 1, 8, 5, 2, 9, 6, 8, 9, 9, 5, 7, 7, 3, 6, 2, 2, 5, 9, 9, 4, 1, 3, 8, 9, 1, 2, 4, 9, 7, 2, 1, 7, 7, 5, 2, 8, 3, 4, 7, 9, 1, 3, 1, 5, 1, 5, 5, 7, 4, 8, 5, 7, 2, 4, 2, 4, 5, 4, 1, 5, 0, 6, 9, 5, 9, 5, 0, 8, 2, 9, 5, 3, 3, 1, 1, 6, 8, 6, 1, 7, 2, 7, 8, 5, 5, 8, 8, 9, 0, 7, 5, 0, 9, 8, 3, 8, 1, 7, 5, 4, 6, 3, 7, 4, 6, 4, 9, 3, 9, 3, 1, 9, 2, 5, 5, 0, 6, 0, 4, 0, 0, 9, 2, 7, 7, 0, 1, 6, 7, 1, 1, 3, 9, 0, 0, 9, 8, 4, 8, 8, 2, 4, 0, 1, 2, 8, 5, 8, 3, 6, 1, 6, 0, 3, 5, 6, 3, 7, 0, 7, 6, 6, 0, 1, 0, 4, 7, 1, 0, 1, 8, 1, 9, 4, 2, 9, 5, 5, 5, 9, 6, 1, 9, 8, 9, 4, 6, 7, 6, 7, 8, 3, 7, 4, 4, 9, 4, 4, 8, 2, 5, 5, 3, 7, 9, 7, 7, 4, 7, 2, 6, 8, 4, 7, 1, 0, 4, 0, 4, 7, 5, 3, 4, 6, 4, 6, 2, 0, 8, 0, 4, 6, 6, 8, 4, 2, 5, 9, 0, 6, 9, 4, 9, 1, 2, 9, 3, 3, 1, 3, 6, 7, 7, 0, 2, 8, 9, 8, 9, 1, 5, 2, 1, 0, 4, 7, 5, 2, 1, 6, 2, 0, 5, 6, 9, 6, 6, 0, 2, 4, 0, 5, 8, 0, 3, 8, 1, 5, 0, 1, 9, 3, 5, 1, 1, 2, 5, 3, 3, 8, 2, 4, 3, 0, 0, 3, 5, 5, 8, 7, 6, 4, 0, 2, 4, 7, 4, 9, 6, 4, 7, 3, 2, 6, 3, 9, 1, 4, 1, 9, 9, 2, 7, 2, 6, 0, 4, 2, 6, 9, 9, 2, 2, 7, 9, 6, 7, 8, 2, 3, 5, 4, 7, 8, 1, 6, 3, 6, 0, 0, 9, 3, 4, 1, 7, 2, 1, 6, 4, 1, 2, 1, 9, 9, 2, 4, 5, 8, 6, 3, 1, 5, 0, 3, 0, 2, 8, 6, 1, 8, 2, 9, 7, 4, 5, 5, 5, 7, 0, 6, 7, 4, 9, 8, 3, 8, 5, 0, 5, 4, 9, 4, 5, 8, 8, 5, 8, 6, 9, 2, 6, 9, 9, 5, 6, 9, 0, 9, 2, 7, 2, 1, 0, 7, 9, 7, 5, 0, 9, 3, 0, 2, 9, 5, 5]

Chapitre 5. Les instructions et fonctions du langage Scilab

ABS

La fonction abs() renvoie la valeur absolue d'un réel ou le module d'un nombre complexe.
Exemple :

```
-->abs(-.35)
 ans  =

    0.35

-->abs(3 + %i*4)
 ans  =

    5.
```

ASCII

La fonction ascii() convertit une chaîne de caractères en codes ascii ou un vecteur de codes ascii en chaîne de caractères. Exemple : ascii("c") renvoie 99 et ascii(99) renvoie "c".

```
Editeur :
u = "ciel";
v = ascii(u);
disp(v)
w = ascii(v);
disp(w)
```

```
Console :
    99.    105.    101.    108.

 ciel
```

CLF

La fonction clf() efface le contenu de la fenêtre graphique courante.
Remarque : On peut saisir dans l'éditeur : clf ou clf(). Dans le cas de plusieurs fenêtres graphiques, l'instruction clf(n) efface le contenu de la fenêtre graphique dont l'identificateur est n.

DEFF

La fonction deff() permet de définir une fonction utilisateur.

```
deff('[z] = mamoyenne(x,y)','z = (x + y)/2')
```

Les paramètres sont deux chaînes de caractères (que l'on définit à l'aide de l'apostrophe ' ou du guillemet ").
x, y : variables d'entrée.
z : variable de sortie.

Remarque : Cette fonction est utilisée pour définir une nouvelle fonction depuis la console Scilab. Cependant elle permet également de définir une fonction au cours d'un programme lorsque l'expression de celle-ci doit être saisie par l'utilisateur.

Exemple :

```
Editeur :
deff('[z] = mamoyenne(x,y)','z = (x + y)/2')
```

```
Console :
mamoyenne(6,8)
 ans  =

    7
```

DISP

La commande disp() permet d'afficher un résultat dans la console Scilab.

```
Editeur :
a = 5;
disp(a)
```

```
Console :
5.
```

EVSTR

Renvoie le résultat de l'évaluation de la matrice de chaînes de caractères. Chaque élément de la matrice doit être une expression Scilab valide.

```
Editeur :
a = 1;
b = 2;
Z = ['a','b'];
mat = evstr(Z);
disp(mat);
```

```
Console :
    1.    2.
```

Chapitre 6. Les instructions et fonctions du langage Python

ABS

Renvoie la valeur absolue d'un réel ou le module d'un nombre complexe.
Exemple :

```
Console :
>>> abs(-5)
5
>>> c = complex(3,4)
>>> abs(c)
5.0
```

a%b

a%b renvoie le reste de la division euclidienne de a par b.
Exemple :

```
Console :
>>> 7%2
1
```

Effectivement : 7 = 2*3 + 1

BREAK

L’instruction break permet de sortir d’une boucle.

Exemple :

```
Editeur :
for i in range(1,1000) :
    print(i)
    if i == 3 :
        break
```

```
Console :
>>>
1
2
3
```

CHR

L’instruction chr() renvoie le caractère correspondant au code ascii.

Exemple :

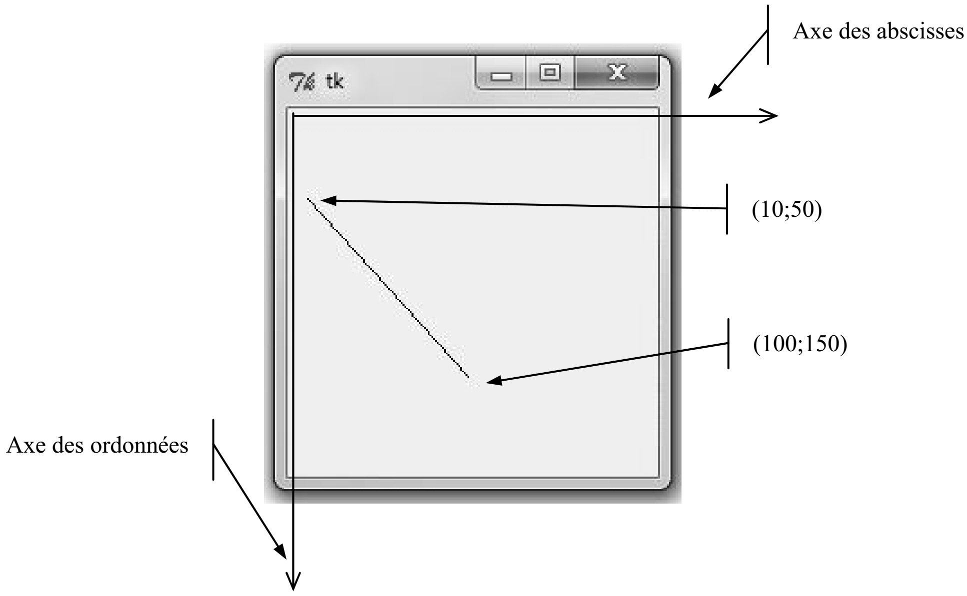
```
Console :
>>> chr(65)
'A'
>>> chr(66)
'B'
```

CREATE_LINE

La fonction create_line(x1,y1,x2,y2) trace le segment de droite [A1A2] avec A1(x1;y1) et A2(x2,y2).

Exemple :

```
Editeur :
from tkinter import *
fen = Tk()
cv = Canvas(fen,width = 200, height = 200)
cv.pack()
cv.create_line(10, 50, 100, 150)
fen.mainloop()
```



Remarque : L'origine du repère se trouve dans le coin supérieur gauche de la fenêtre.

DEF

Déclaration d'une fonction personnalisée.

Exemple :

```
Editeur :
def f(x) :
    y = x**2 + 1
    return y
```

Par exemple, f(2) revoie 5.