

☐

I'm not robot


reCAPTCHA

Continue

Exercice multiplication a virgule 6eme pdf

D'écéption, notre tentative échoue : on rem arq ue que la valeur 3 a p u r ta n b e l et en est ée récupérée p ar le program m e et que la variable x sem ble co n t e n i r l'en tier 3. Or il n' "est rien. Ce que contient la variable x, c'est la chaîne de caractères "3". Et u n e chaîne de caractères ne peut être élevée au carré. D'où la nécessité de parler t o t u t du type d' u n objet... Sans trop chahuter dans les détails du fo n ctionnem ent d'un ordinateur, rappelons que t u e information ation, et en particulier le co n t e n u d' u n e variable, doit étre codée en binaire. Mais le "dictionnaire" p o u r coder ou décoder le co n t e n u d' u n e variable ne sera pas le m ê m e suivant que l'on a affaire à u n e n ier, à u n o m bre ou natio n scientifique ou u n e chaîne de caractères. Dans ces conditions, la connaissance du co n t e n u d' u n e variable ne suffit pas p o u r d éterm iner l'infor m a tion correspondante. Il est nécessaire de savoir, en outre, co m m e n t la valeur qui s'y trouve est codée. Cette distinction correspond à la no t i o n de type. Pour connaître le type d' u n objet, nous avons déjà vu q u' "on dispose de la fonction ty p e > ; ty p e (3), ty p e (" 3 "), ty p e (x) ("< class str">, "< class str">") Ainsi 3 est du type entier (i n t étan t l'abréviation du mo t "integer"), tandis que "3" est du type chaîne de caractères (s t r étan t l'abréviation du mo t "string").

Il y a donc une différence entre la notation d'expression lambda `x = 3` et la notation `lambda x: x**2` (<http://www.biolbio.net/tutoriels/lambda.html>) car la première crée une fonction qui retourne le carré de son argument, tandis que la seconde crée une fonction qui retourne la notation littérale `"9"`.

Le langage de programmation Python pour les mathématiques n'a pas un peu dans le m ê m e ordre d'idées : la fonction `eval()` qui perm et d'exécuter u n code en s t r étan t représentée par u n e chaîne de caractères. Donnons u n petit exm ple sans le détailler (pour se faire un peu plus sûr, il faille appeler le m o d u le math qui perm et d'im p o rter les fonctions m ath ématiq ues usuelles, ainsi que certaines constantes usuelles com m e le nombre d'Euclide, tan h, from math i m p o r t *X*= eval (in put ("x"= ?")) co n t i o n = in put ("f"= ?") code = ("def f(x) : return (x ** f.co n t i o n)") exec(code) print ("f(6.7)", f.o r m a t (f(f(x)))-2)= 2pi/4 f= 2*pi sin (x) 0.707107 Pour d'm p rimer cette section, où nous avons appris à écrire de petits scripts dans des fichiers, signalons q u'il est bon de prendre t u t de la suite l'h ab itud e de bien p résen ter son code en Py thon selon les conventions suivantes : - taille des indentations : 4 espaces ; - taille m axim ale à u n e ligne : 79 caractères ; - toujours placer u n espace après une virgule, u n point-virgule ou deux-points (sauf p o u r la syntaxe des tranches) ; - ne jam ais placer d' "espace avant un e virgule, u n point-virgule ou deux-points ; - toujours placer un espace de chaque côté d' u n opérateur ; - ne pas placer d' "espace entre le nom d' u n e fonction et sa liste d'arguments.

Nom : _____ Date : ____/____/____

Multiplier et diviser par 10, 100, 0.1, 0.01

Exercice 1 : Multiplier et diviser par 10, 100, 1000.

Compléter par le nombre qui convient.

a 19.5 X = 1950 b 1.086 X = 1086
c 0.03 X = 0.3 d 14 + = 0.14
e 563 + = 56.3 f 1236 + = 1.236

Exercice 2 : Par 10, 100 ou 1000.

Compléter les égalités suivantes en multipliant par 10, 100 ou 1000 pour obtenir le plus petit entier possible.

a 7.63 X =
b 89.1 X =
c 0.013 X =
d 0.01 X =
e 13.823 X =
f 1.27 X =

Exercice 3 : la pyramide

Compléter la pyramide suivante sachant qu'une brique est égale au produit des deux briques qui la soutiennent.

Exercice 4 : le mètre

Un ticket de métro coûte 1.50 euros à l'unité. Un carnet de 10 tickets coûte 13.35 euros. Combien économise-t-on en prenant un carnet plutôt que d'acheter 10 tickets à l'unité ?

Exercice 5 : Le message caché.

a Donner le résultat sans poser l'opération, notez dans la première colonne.
b Remplir la troisième colonne en donnant le chiffre demandé.
c Pour découvrir le message indiqué, dans la dernière colonne, la lettre correspondant au code ci-dessous.

0	1	2	3	4	5	6	7	8
F	R	I	S	E	V	A	O	D

1.2 × 0.01 =	Chiffre des unités	
693 X 0.1 =	Chiffre des dizaines	
26.3 × 10 =	Chiffre des unités	
51.5 × 100 =	Chiffre des centimes	
15.94 X 100 =	Chiffre des unités	
60.03 × 10 =	Chiffre des millièmes	
0.14 X 10 =	Chiffre des dizaines	
1368 × 10 =	Chiffre des dizaines	
8 × 0.01 =	Chiffre des centimes	
15.4 X 10 =	Chiffre des unités	
52 X 0.001 =	Chiffre des centimes	
0.7 X 1000 =	Chiffre des centaines	
200 = 10 =	Chiffre des dizaines	
13 X 10 =	Chiffre des centimes	
3 = 1000 =	Chiffre des millièmes	

Message :

www.passe-education.fr

[illegible]

3

Date _____

Complète les opérations de multiplication en te servant de la méthode classique.

5 6 8

×

1

5 7 5

×

7

6 8 9

×

4

6 3 2

×

3

6 3 3

×

9

5 2 7

×

2

6 4 7

×

6

5 5 5

×

8

5 9 7

×

7

6 1 7

×

3

5 2 9

×

9

6 1 9

×

2

6 3 7

×

6

6 6 7

×

5

5 8 7

×

4

5 4 7

×

5

A ttardons-nous quelques instants sur les booléens. Pour ob ten ir u n e variable booléenne, on utilise en général, com m e d ans l'exem ple précédent, u n ou plusieurs opérateurs de com parai son. Ces opérateurs sont au nom bre de h u it : "O c n Û U D tH o rs l (S) oi > .Q.

[illegible][illegible]

[illegible]

[illegible]

122 2.4 Réels et flottants : un très rapide survol de la norme IEEE754.....	132 2.5 Les nombres complexes.....	142 2.6 Exercices d'entraînement.....	146 Le nombre n	149 3.1 L'approximation de j de
Nicolas D e C u e s.....	149 3.2 Approximation de π et calcul approché d'intégrale.....	151 3.3 Le nombre 7π et les arctangentes.....	156 3.4 Méthodes modernes.....	161 3.5 Algorithme de P l o u f f e
	162 Probabilités.....	163 4.1 Pile ou face ?.....	164 4.2 Tirage de boules dans des urnes.....	164 4.3 Tirage de boules avec remise
	165 4.4 Problème du Duc de To scan e.....	165 4.5 Lancers d'une pièce et résultats égaux.....	167 4.6 Le Monty H a l l.....	168 4.7 Exercice d'entraînement

170.5.1. Relations binaires sur un ensemble 170.5.2. Graphes 173

www.biblio-scientifique.net Programmation en Python pour les mathématiques 78 | Matrices 1.1 Les matrices avec NumPy et SciPy Le module `scipy.linalg` propose tous les outils nécessaires au calcul matriciel et utilise la bibliothèque NumPy ; les opérations arithmétiques basiques ainsi qu'une soixantaine de méthodes couvrant les besoins standards sont à la disposition des utilisateurs.

```
>>> >>> import numpy as np from scipy.linalg import A = np.array([(1, 2), (3, 4)]) A array([[1, 2], [3, 4]]) >>> A * 3 # chaque terme est multiplié par 3 array([[3, 6], [9, 12]]) >>> A + 10 * array([[1, 1], [13, 14]]) # chaque terme est additionné à 10 array([[11, 14], [17, 22]]) Attention ! l'opérateur * effectue le produit entre termes. Pour multiplier deux matrices utilisez dot : >>> A * array([(1, 9, 4), A**2 * array([(1, 9, 16)], >>> A.dot(A) array([(7, 10], [15, 22)]) On peut calculer le déterminant et l'inverse d'une matrice facilement : >>> A.O = 3 * J.D.T.H.O.C.M >>> 1/A # c'est l'inverse terme à terme array([(1, -1), [-0.5, [0.33333333, 0.25]]) >>> iA = linalg.pinv(A)
```

[illegible][illegible][illegible]

ont tous les caractères e x i s t a n t s . L a p l u p a r t d e s l a n g a g e s d e p r o g r a m m a t i o n , e t i e n s u n p r y p t o n , o n t u n e m a n d e q u i a s s o c i e l e c a r a c t è r e A C I I (e n t r e e l l e m e n t e n t è n d u) à u n c a r a c t è r e . O n p o u r r a d o n c L u t i l i s e r p o u r r é s o u d r e n o t r e p r o b l è m e l ' e n s a g i s s a n t d e m a n d e o r d (c a r a t è r e) q u i r e n v o i e l e c i d e A C I I d u n c a r a c t è r e c a r a c t è r e . C a r a c t è r e C o d e C a r a c t è r e C a r a c t è r e C a r a c t è r e C o d e ■ 3 4 5 5 3 F 7 8 7 0 7 1 7 2 7 1 [8 9 9 9 9 1 1 m n 1 0 8 1 0 9 1 1 0 1 3 3 4 5 2 3 2 3 5 1 f a t 5 6 5 4 3 5 6 7 5 5 3 7 8 5 6 7 5 3 8 5 7 1 7 6 ■ 3 9 / 4 0 : 5 8 5 9 I M N J 7 3 7 4 7 7 7 8 * a 1 9 2 9 3 4 9 5 9 6 9 7 P o s t j 1 1 1 1 1 2 1 1 3 1 1 4 1 1 5 1 1 1 6 1 4 1 4 0 6 7 9 b 9 8 U 1 1 7 - * 4 2 = 6 1 B 6 r 0 9 9 V 1 1 8 + 4 3 = 6 2 Q 4 1 d 1 0 0 w 1 1 9 y 4 4 = 6 3 R 8 2 e 1 0 1 X 1 2 4 5 6 7 8 9 1 0 1 1 2 1 2 1 4 6 A 6 5 T 8 4 q 1 2 3 0 4 1 2 2 7 4 4 6 B 6 U 5 8 5 1 0 4 [1 2 3 0 4 8 6 7 8 6 8 1 0 5 1 2 4 1 4 9 D 6 8 W 8 7 j 1 0 6 j 1 2 5 2 5 6 9 4 8 X 8 8 k 1 0 7 1 2 6 T a b l e 3 . 1 . T a b l e d e s c a r a c t è r e s A C I I a f f i c h a b l e s P a r e x e m p l e : > > o r d (' a / 6 4 M 5 a t t e n t i o n C l e a n e p o u r c n n e p a r o l e s c h a i n e s d e c a r a c t è r e s : > > o r d (' A B T r a c e b o o t m o c e n t c a l l a s t) : f l e q < s t d i n > , l i n e l , i n T y p e R e p o r t (o r d) e x p e c t e d a c h a r a c t e r , b u t s t r i n g o f l e n g t h 2 f o u n d I f o u r d a r o n c d o n s i d è r e l e c a r a c t è r e s d ' u n c h a i n e a n p a r u n : T J C o r ç j u O I H O T M (y) a n : > > O C L U d e f a s s i c i o n e : r e t u r n l i n e (c o d e c a r a c t è r e d e l ' o r d i n e p e r m e t d ' a f f i c h e r l e s c a r a c t è r e s c o d é s) ■ 3 4 5 5 3 F 7 8 7 0 7 1 7 2 7 1 [8 9 9 9 9 1 1 m n 1 0 8 1 0 9 1 1 0 1 3 3 4 5 2 3 2 3 5 1 f a t 5 6 5 4 3 5 6 7 5 5 3 7 8 5 6 7 5 3 8 5 7 1 7 6 ■ 3 9 / 4 0 : 5 8 5 9 I M N J 7 3 7 4 7 7 7 8 * a 1 9 2 9 3 4 9 5 9 6 9 7 P o s t j 1 1 1 1 1 2 1 1 3 1 1 4 1 1 5 1 1 1 6 1 4 1 4 0 6 7 9 b 9 8 U 1 1 7 - * 4 2 = 6 1 B 6 r 0 9 9 V 1 1 8 + 4 3 = 6 2 Q 4 1 d 1 0 0 w 1 1 9 y 4 4 = 6 3 R 8 2 e 1 0 1 X 1 2 4 5 6 7 8 9 1 0 1 1 2 1 2 1 4 6 A 6 5 T 8 4 q 1 2 3 0 4 1 2 2 7 4 4 6 B 6 U 5 8 5 1 0 4 [1 2 3 0 4 8 6 7 8 6 8 1 0 5 1 2 4 1 4 9 D 6 8 W 8 7 j 1 0 6 j 1 2 5 2 5 6 9 4 8 X 8 8 k 1 0 7 1 2 6 T a b l e 3 . 1 . T a b l e d e s c a r a c t è r e s A C I I a f f i c h a b l e s P a r e x e m p l e : > > o r d (' a / 6 4 M 5 a t t e n t i o n C l e a n e p o u r c n n e p a r o l e s c h a i n e s d e c a r a c t è r e s : > > o r d (' A B T r a c e b o o t m o c e n t c a l l a s t) : f l e q < s t d i n > , l i n e l , i n T y p e R e p o r t (o r d) e x p e c t e d a c h a r a c t e r , b u t s t r i n g o f l e n g t h 2 f o u n d I f o u r d a r o n c d o n s i d è r e l e c a r a c t è r e s d ' u n c h a i n e a n p a r u n : T J C o r ç j u O I H O T M (y) a n : > > O C L U d e f a s s i c i o n e : r e t u r n l i n e (c o d e c a r a c t è r e d e l ' o r d i n e p e r m e t d ' a f f i c h e r l e s c a r a c t è r e s c o d é s) ■ 3 4 5 5 3 F 7 8 7 0 7 1 7 2 7 1 [8 9 9 9 9 1 1 m n 1 0 8 1 0 9 1 1 0 1 3 3 4 5 2 3 2 3 5 1 f a t 5 6 5 4 3 5 6 7 5 5 3 7 8 5 6 7 5 3 8 5 7 1 7 6 ■ 3 9 / 4 0 : 5 8 5 9 I M N J 7 3 7 4 7 7 7 8 * a 1 9 2 9 3 4 9 5 9 6 9 7 P o s t j 1 1 1 1 1 2 1 1 3 1 1 4 1 1 5 1 1 1 6 1 4 1 4 0 6 7 9 b 9 8 U 1 1 7 - * 4 2 = 6 1 B 6 r 0 9 9 V 1 1 8 + 4 3 = 6 2 Q 4 1 d 1 0 0 w 1 1 9 y 4 4 = 6 3 R 8 2 e 1 0 1 X 1 2 4 5 6 7 8 9 1 0 1 1 2 1 2 1 4 6 A 6 5 T 8 4 q 1 2 3 0 4 1 2 2 7 4 4 6 B 6 U 5 8 5 1 0 4 [1 2 3 0 4 8 6 7 8 6 8 1 0 5 1 2 4 1 4 9 D 6 8 W 8 7 j 1 0 6 j 1 2 5 2 5 6 9 4 8 X 8 8 k 1 0 7 1 2 6 T a b l e 3 . 1 . T a b l e d e s c a r a c t è r e s A C I I a f f i c h a b l e s P a r e x e m p l e : > > o r d (' a / 6 4 M 5 a t t e n t i o n C l e a n e p o u r c n n e p a r o l e s c h a i n e s d e c a r a c t è r e s : > > o r d (' A B T r a c e b o o t m o c e n t c a l l a s t) : f l e q < s t d i n > , l i n e l , i n T y p e R e p o r t (o r d) e x p e c t e d a c h a r a c t e r , b u t s t r i n g o f l e n g t h 2 f o u n d I f o u r d a r o n c d o n s i d è r e l e c a r a c t è r e s d ' u n c h a i n e a n p a r u n : T J C o r ç j u O I H O T M (y) a n : > > O C L U d e f a s s i c i o n e : r e t u r n l i n e (c o d e c a r a c t è r e d e l ' o r d i n e p e r m e t d ' a f f i c h e r l e s c a r a c t è r e s c o d é s) ■ 3 4 5 5 3 F 7 8 7 0 7 1 7 2 7 1 [8 9 9 9 9 1 1 m n 1 0 8 1 0 9 1 1 0 1 3 3 4 5 2 3 2 3 5 1 f a t 5 6 5 4 3 5 6 7 5 5 3 7 8 5 6 7 5 3 8 5 7 1 7 6 ■ 3 9 / 4 0 : 5 8 5 9 I M N J 7 3 7 4 7 7 7 8 * a 1 9 2 9 3 4 9 5 9 6 9 7 P o s t j 1 1 1 1 1 2 1 1 3 1 1 4 1 1 5 1 1 1 6 1 4 1 4 0 6 7 9 b 9 8 U 1 1 7 - * 4 2 = 6 1 B 6 r 0 9 9 V 1 1 8 + 4 3 = 6 2 Q 4 1 d 1 0 0 w 1 1 9 y 4 4 = 6 3 R 8 2 e 1 0 1 X 1 2 4 5 6 7 8 9 1 0 1 1 2 1 2 1 4 6 A 6 5 T 8 4 q 1 2 3 0 4 1 2 2 7 4 4 6 B 6 U 5 8 5 1 0 4 [1 2 3 0 4 8 6 7 8 6 8 1 0 5 1 2 4 1 4 9 D 6 8 W 8 7 j 1 0 6 j 1 2 5 2 5 6 9 4 8 X 8 8 k 1 0 7 1 2 6 T a b l e 3 . 1 . T a b l e d e s c a r a c t è r e s A C I I a f f i c h a b l e s P a r e x e m p l e : > > o r d (' a / 6 4 M 5 a t t e n t i o n C l e a n e p o u r c n n e p a r o l e s c h a i n e s d e c a r a c t è r e s : > > o r d (' A B T r a c e b o o t m o c e n t c a l l a s t) : f l e q < s t d i n > , l i n e l , i n T y p e R e p o r t (o r d) e x p e c t e d a c h a r a c t e r , b u t s t r i n g o f l e n g t h 2 f o u n d I f o u r d a r o n c d o n s i d è r e l e c a r a c t è r e s d ' u n c h a i n e a n p a r u n : T J C o r ç j u O I H O T M (y) a n : > > O C L U d e f a s s i c i o n e : r e t u r n l i n e (c o d e c a r a c t è r e d e l ' o r d i n e p e r m e t d ' a f f i c h e r l e s c a r a c t è r e s c o d é s) ■ 3 4 5 5 3 F

[illegible]

ue l'ont travaillé en base 2 : notre nombre, qui n'est pas un flottant de Y64, est à équidistance de 1,000...000 et 1,000...001. Il se rajoute à ce lui qui se termine en $V^{\wedge}-----V-----$ '52bits 52bits par un $0:2^{\wedge}^{\wedge}$. $RN(2^{\wedge}^{\wedge}+1)=2^{\wedge}^{\wedge}$ Vous pouvez alors comprendre pourquoi : $RN(2^{\wedge}^{\wedge}+3)=$

[illegible]

196) p ou p et permet la vectorisation des données : la liste des points calculés doit être présent u tableau du type numpy . array (cste p / 74).

a) Pour tester cette nouvelle classe ODE, tracer les courbes de l'oscillateur harmonique simple en python avec m a t r i c l e b l o k s les courbes intégrales du système a u t o n o m e d e v a n d e r P o l e n utilisant non plus la méthode RK4 (système m ais RK4). Les exercices suivants se proposent de reprendre quelques schém a m on ériques présentés dans ce chapitre en utilisant cette fonction-ci les primitives disponibles dans le module S c i p y (Point fixe du cosinus) En im p o r t a n t la fonction o p t i m i z e d u m o d u l e S c i p y , trouver la valeur du point f i x e de la fonction cosinus, c-à-d. la solution de l'équation $\cos x = x$. E x e r c i c e 5 2 . S o l u t i o n . >>> >>> Pour des variantes de la solution ci-dessous, consulter le fichier de corrigés en ligne. Im p o r t numpy as np from scipy import optimize, bisect def f(x): return -cos(x)-x, 0 #règle de Newton 0.7390851332156672 E x e r c i c e 5 3 . (Intégration num érique) Utiliser la fonction integrate du module S c i p y pour trouver une approximation de $\int_0^1 |t| dt = 4 - \int_0^1 |V(t)|^2 dt$ et $\int_0^1 |t|^2 dt = 12$ à l'aide de V(1)=1 et de t(1)=C par les résultats avec la valeur de C fournie par SciPy. (Équations différentielles) En im p o r t a n t la fonction integrate du module S c i p y , tracer les solutions des équations différentielles suivantes rencontrées à la section 3 : E x e r c i c e 5 4 .

T3 O C 3 Q k o r M (5) $t(x) = (1-x)^{-\frac{1}{2}} \cdot x^{(\frac{1}{2})} B = y(x) t'(x) = 0$ $x t(x) + (1-x) t'(x) = 0$ $H x x C T I > Q$.

Pour l'installation et la documentation du module S c i p y , consulter h t t p : // s c i p y . o r g / , w w w . b i b l i o - s c i e n t i f i q u e . n e t R é c u r s i v e Sommaire 1 Quelques exemples 1 PGCD d'une liste 217 1. 2 Compléments sur la récursivité 3 Complexité..... 230 Sommes de termes d'une suite géométrique 234 2. 3 Construction récursive d'un arbre de recherche 224 2. 4 La construction récursive d'un arbre de recherche 224 2. 5 Les arbres de recherche 224 2. 6 Les arbres de recherche 224 2. 7 Les arbres de recherche 224 2. 8 Les arbres de recherche 224 2. 9 Les arbres de recherche 224 2. 10 Les arbres de recherche 224 2. 11 Les arbres de recherche 224 2. 12 Les arbres de recherche 224 2. 13 Les arbres de recherche 224 2. 14 Les arbres de recherche 224 2. 15 Les arbres de recherche 224 2. 16 Les arbres de recherche 224 2. 17 Les arbres de recherche 224 2. 18 Les arbres de recherche 224 2. 19 Les arbres de recherche 224 2. 20 Les arbres de recherche 224 2. 21 Les arbres de recherche 224 2. 22 Les arbres de recherche 224 2. 23 Les arbres de recherche 224 2. 24 Les arbres de recherche 224 2. 25 Les arbres de recherche 224 2. 26 Les arbres de recherche 224 2. 27 Les arbres de recherche 224 2. 28 Les arbres de recherche 224 2. 29 Les arbres de recherche 224 2. 30 Les arbres de recherche 224 2. 31 Les arbres de recherche 224 2. 32 Les arbres de recherche 224 2. 33 Les arbres de recherche 224 2. 34 Les arbres de recherche 224 2. 35 Les arbres de recherche 224 2. 36 Les arbres de recherche 224 2. 37 Les arbres de recherche 224 2. 38 Les arbres de recherche 224 2. 39 Les arbres de recherche 224 2. 40 Les arbres de recherche 224 2. 41 Les arbres de recherche 224 2. 42 Les arbres de recherche 224 2. 43 Les arbres de recherche 224 2. 44 Les arbres de recherche 224 2. 45 Les arbres de recherche 224 2. 46 Les arbres de recherche 224 2. 47 Les arbres de recherche 224 2. 48 Les arbres de recherche 224 2. 49 Les arbres de recherche 224 2. 50 Les arbres de recherche 224 2. 51 Les arbres de recherche 224 2. 52 Les arbres de recherche 224 2. 53 Les arbres de recherche 224 2. 54 Les arbres de recherche 224 2. 55 Les arbres de recherche 224 2. 56 Les arbres de recherche 224 2. 57 Les arbres de recherche 224 2. 58 Les arbres de recherche 224 2. 59 Les arbres de recherche 224 2. 60 Les arbres de recherche 224 2. 61 Les arbres de recherche 224 2. 62 Les arbres de recherche 224 2. 63 Les arbres de recherche 224 2. 64 Les arbres de recherche 224 2. 65 Les arbres de recherche 224 2. 66 Les arbres de recherche 224 2. 67 Les arbres de recherche 224 2. 68 Les arbres de recherche 224 2. 69 Les arbres de recherche 224 2. 70 Les arbres de recherche 224 2. 71 Les arbres de recherche 224 2. 72 Les arbres de recherche 224 2. 73 Les arbres de recherche 224 2. 74 Les arbres de recherche 224 2. 75 Les arbres de recherche 224 2. 76 Les arbres de recherche 224 2. 77 Les arbres de recherche 224 2. 78 Les arbres de recherche 224 2. 79 Les arbres de recherche 224 2. 80 Les arbres de recherche 224 2. 81 Les arbres de recherche 224 2. 82 Les arbres de recherche 224 2. 83 Les arbres de recherche 224 2. 84 Les arbres de recherche 224 2. 85 Les arbres de recherche 224 2. 86 Les arbres de recherche 224 2. 87 Les arbres de recherche 224 2. 88 Les arbres de recherche 224 2. 89 Les arbres de recherche 224 2. 90 Les arbres de recherche 224 2. 91 Les arbres de recherche 224 2. 92 Les arbres de recherche 224 2. 93 Les arbres de recherche 224 2. 94 Les arbres de recherche 224 2. 95 Les arbres de recherche 224 2. 96 Les arbres de recherche 224 2. 97 Les arbres de recherche 224 2. 98 Les arbres de recherche 224 2. 99 Les arbres de recherche 224 2. 100 Les arbres de recherche 224 2. 101 Les arbres de recherche 224 2. 102 Les arbres de recherche 224 2. 103 Les arbres de recherche 224 2. 104 Les arbres de recherche 224 2. 105 Les arbres de recherche 224 2. 106 Les arbres de recherche 224 2. 107 Les arbres de recherche 224 2. 108 Les arbres de recherche 224 2. 109 Les arbres de recherche 224 2. 110 Les arbres de recherche 224 2. 111 Les arbres de recherche 224 2. 112 Les arbres de recherche 224 2. 113 Les arbres de recherche 224 2. 114 Les arbres de recherche 224 2. 115 Les arbres de recherche 224 2. 116 Les arbres de recherche 224 2. 117 Les arbres de recherche 224 2. 118 Les arbres de recherche 224 2. 119 Les arbres de recherche 224 2. 120 Les arbres de recherche 224 2. 121 Les arbres de recherche 224 2. 122 Les arbres de recherche 224 2. 123 Les arbres de recherche 224 2. 124 Les arbres de recherche 224 2. 125 Les arbres de recherche 224 2. 126 Les arbres de recherche 224 2. 127 Les arbres de recherche 224 2. 128 Les arbres de recherche 224 2. 129 Les arbres de recherche 224 2. 130 Les arbres de recherche 224 2. 131 Les arbres de recherche 224 2. 132 Les arbres de recherche 224 2. 133 Les arbres de recherche 224 2. 134 Les arbres de recherche 224 2. 135 Les arbres de recherche 224 2. 136 Les arbres de recherche 224 2. 137 Les arbres de recherche 224 2. 138 Les arbres de recherche 224 2. 139 Les arbres de recherche 224 2. 140 Les arbres de recherche 224 2. 141 Les arbres de recherche 224 2. 142 Les arbres de recherche 224 2. 143 Les arbres de recherche 224 2. 144 Les arbres de recherche 224 2. 145 Les arbres de recherche 224 2. 146 Les arbres de recherche 224 2. 147 Les arbres de recherche 224 2. 148 Les arbres de recherche 224 2. 149 Les arbres de recherche 224 2. 150 Les arbres de recherche 224 2. 151 Les arbres de recherche 224 2. 152 Les arbres de recherche 224 2. 153 Les arbres de recherche 224 2. 154 Les arbres de recherche 224 2. 155 Les arbres de recherche 224 2. 156 Les arbres de recherche 224 2. 157 Les arbres de recherche 224 2. 158 Les arbres de recherche 224 2. 159 Les arbres de recherche 224 2. 160 Les arbres de recherche 224 2. 161 Les arbres de recherche 224 2. 162 Les arbres de recherche 224 2. 163 Les arbres de recherche 224 2. 164 Les arbres de recherche 224 2. 165 Les arbres de recherche 224 2. 166 Les arbres de recherche 224 2. 167 Les arbres de recherche 224 2. 168 Les arbres de recherche 224 2. 169 Les arbres de recherche 224 2. 170 Les arbres de recherche 224 2. 171 Les arbres de recherche 224 2. 172 Les arbres de recherche 224 2. 173 Les arbres de recherche 224 2. 174 Les arbres de recherche 224 2. 175 Les arbres de recherche 224 2. 176 Les arbres de recherche 224 2. 177 Les arbres de recherche 224 2. 178 Les arbres de recherche 224 2. 179 Les arbres de recherche 224 2. 180 Les arbres de recherche 224 2. 181 Les arbres de recherche 224 2. 182 Les arbres de recherche 224 2. 183 Les arbres de recherche 224 2. 184 Les arbres de recherche 224 2. 185 Les arbres de recherche 224 2. 186 Les arbres de recherche 224 2. 187 Les arbres de recherche 224 2. 188 Les arbres de recherche 224 2. 189 Les arbres de recherche 224 2. 190 Les arbres de recherche 224 2. 191 Les arbres de recherche 224 2. 192 Les arbres de recherche 224 2. 193 Les arbres de recherche 224 2. 194 Les arbres de recherche 224 2. 195 Les arbres de recherche 224 2. 196 Les arbres de recherche 224 2. 197 Les arbres de recherche 224 2. 198 Les arbres de recherche 224 2. 199 Les arbres de recherche 224 2. 200 Les arbres de recherche 224 2. 201 Les arbres de recherche 224 2. 202 Les arbres de recherche 224 2. 203 Les arbres de recherche 224 2. 204 Les arbres de recherche 224 2. 205 Les arbres de recherche 224 2. 206 Les arbres de recherche 224 2. 207 Les arbres de recherche 224 2. 208 Les arbres de recherche 224 2. 209 Les arbres de recherche 224 2. 210 Les arbres de recherche 224 2. 211 Les arbres de recherche 224 2. 212 Les arbres de recherche 224 2. 213 Les arbres de recherche 224 2. 214 Les arbres de recherche 224 2. 215 Les arbres de recherche 224 2. 216 Les arbres de recherche 224 2. 217 Les arbres de recherche

[illegible][illegible]

And, du moment que la question du coût d'un n est stricto sensu plus élevée selon l'environnement et le giciel, on ne peut pas évaluer la complexité en termes de « instructions élémentaires ». C'est une tautologie. On ne peut pas dire que la complexité en termes de « instructions élémentaires » est une fonction de n dans p ou d de a ou b de c de e de f de g de h de i de j de k de l de m de n de o de p de q de r de s de t de u de v de w de x de y de z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M de N de O de P de Q de R de S de T de U de V de W de X de Y de Z de A de B de C de D de E de F de G de H de I de J de K de L de M

```
a> from CL0_0.py.biblio-scientifique.net226.Programmation en Python pour les mathématiques Solution .pentagones ,py from math import *from turtle import *def pentagone(cote):for nrange(5):forward(cote)left(72)despense(nombre_triangles,cote):coeff=0.1a=coted=coeff*a+for nrange(nombre_triangles):pentagone(a=d=coeff*a+forward(d)a.prime=sqrt((a-d)**2+a**2-2*(a-d)*(d+sqrt(5))/4))angle=(180/pi)/left(angle)a=a-prime+coeff*(a-prime**2-2*a*d)/(2*a-prime+(a-d))up(goto(-190,-210)down)spirale:(50,370)up[3]CourbedudragonmaLacourbedudragonestuneautreapplication delarécursivité.=Qdragon.pywithHofNFJ(SX:xi>Q.Oufrom math importsqrtfrom turtleimport*defdragon_gauche(taille,niveau):color("red")ifniveau==0:forward(taille)else:left(45)dragon_gauche(taille/sqrt(2),niveau-1)right(90)dragon_droite(taille/sqrt(2),niveau-1)left(45)ifniveau==0:forward(taille)else:right(45)dragon_gauche(taille/sqrt(2),left(90)dragon_droite(taille/sqrt(2),right(45)dragon_droite(400,niveau-1)niveau-1)12)Lacourbedudragon paveleplan,comme lemontrelafiguresuivantedragon2.pyfrom math importsqrtfrom turtleimport*TJOcrjüJDÖtHCMY(XI)Oj>
```

[illegible]

```
server(N,Cetalgrouplem,donnelesfiguresuivante,réaliseavecleprogrammelesroutinespinsky.py,Ce(y)programmefaitappelaunouvelautotypeobjetdéfiniparleprogrammeurpoureprésentera).Q.Ounobjetcpointduplan,repéréparseccoordonnéescar-  
tésiennesetdotéd'uncoordonneur. Quelquequesoient, sanscompromettrelessegments[M/M+1]uniquementilesàlaconstructionmaisquiencombrentledessin.
```

[illegible][illegible]

238 1.1	Présentation	239 1.3	Parcours éulériens	240	Représentation des nombres
238 1.2	Quelques questions usuelles	242 2.2	Une première tentative et un échec	246 2.3	Une seconde tentative
238 2.1	Quelques rappels et outils				

247 2.4 Une classe pour les entiers.....			
248 2.5 Une classe pour les rationnels.....	252	Listes	
		257 3.1 Trois versions du tri récursif d'une liste.....	257 3.2 Une classe pour encapsuler l'information.....
			260 3.3 Une classe pour la structure de données.....

[illegible]

On manipule des informations, dont l'intérêt est de donner un cadre général et de transformer une collection en un nombre d'éléments. L'unité de collection ne peut pas être conservée, à l'utilisation du programme, ou d'un problème à résoudre, à la construction d'un programme. Les objets d'une telle collection peuvent être de nature hétérogène ou non, comme ils peuvent être organisés ou non, selon des critères pertinents de finis par le programme pour les résoudre ou pour les traiter. Nous traiterons dans

En la práctica, el desarrollo de los sistemas de gestión de la información se ha convertido en una tarea compleja y multidisciplinaria. En este sentido, el presente artículo tiene como objetivo principal analizar el impacto de la tecnología en la gestión de la información y su relación con la productividad y la competitividad de las organizaciones. Para ello, se ha realizado una revisión bibliográfica de los últimos años, buscando identificar las tendencias y los desafíos más relevantes en este campo. Los resultados indican que la tecnología ha permitido la creación de sistemas más eficientes y seguros, lo que ha contribuido a mejorar la toma de decisiones y la gestión de los recursos. Sin embargo, también se han identificado algunos desafíos, como la falta de capacitación del personal y la seguridad de la información. En conclusión, se puede afirmar que la tecnología es un factor clave para el éxito de las organizaciones en el mundo actual, pero su implementación debe ser cuidadosa y planificada.

Insatidondeshistes.Toutefois,laprésentationdeslistesconstitueunpartimportantdel'algorithmique,etilfautbondeconnaîtrelesfonctionnementinterne decettypeclassique destructurededonnées.Enfin,l'étudedeslistespermettradedeuxautresapproches:celledesalgorithmesde tri,etcelledesalgorithmesde recherche. En effet, les listes sont des structures de données fondamentales pour la programmation, et leur étude permet de mieux comprendre les algorithmes de tri et de recherche, et de mieux comprendre les algorithmes de tri et de recherche.

et de maintenance. Lors d'une conception dans des classes, nous aurons l'occasion d'examiner et de mettre en vue les notions de composition et d'héritage. Nous prendrons comme exemple directeur de cette représentation la réalisation d'un petit système médical appelé «les ombres chinoises» que nous avons écrit en langage C++.

ils ont écrit une méthode pour intégrer des nouvelles fonctionnalités de langage. En particulier, l'élément `urn:nal:cas:conservation` de l'instruction `<print>` est une étiquette de la classe `urn:nal:cas:conservation:print()`. `urn:nal:cas:conservation:print()` est une instance de la classe `urn:nal:cas:conservation:print()`. En fin de compte, dans un but d'étude, nous n'avons pas cherché à utiliser toutes les potentialités du langage.

minutives puis parfois de fait « opaques » — a été le cours ou plutôt la transférence des concepts présents vers une autre langue de programmation. 1. Graphes. Avant de nous pencher dans les sections suivantes sur les notions de arbres et de listes qui nous ne devons pas applications, nous commençons ce chapitre en abordant la notion de graphe, qui permet notamment de généraliser la notion de deux notions présentes, au sens où nous abordons un graphe comme un cas particulier d'un graphe connexe sans cycle. Si la théorie des graphes constitue une

unilatérale d'adjacence. Nous choisissons aussi des exemples de treillis de graphes non orientés aux arêtes non pondérées (1.1) et (2.1) (Cf. [9]). Dans l'annexe D, nous présentons deux types de représentation du contour à l'aide d'adjacence unilatérale d'adjacence. Une matriced'adjacence d'adjacence unilatérale symétrique (pour un graphe non orienté), dont les éléments (dans notre cas, égaux à des booléens) indiquent l'existence d'un arc entre deux sommets. CML programme ci

[illegible]

[A, A', A'', A''', D, D', D'', D''', B', B'', B''', C, C'', C''', s], s) OcrDpKDmO B', B'', B''', E', E'', E''', Terminal]

de l'autre, la constitution d'un excellent exercice de programmation confiée à l'élève. 1.2. Quelques questions usuelles Nous nous proposons d'exercer les compétences de programmation – les questions suivantes : Un graphe est-il connexe ? www.biblio-scientifique.net 240 Programmation en Python pour les mathématiciens A101010000B110111100C1100001000D1110011000E1011010100F101101000G100000001H1010000010J A—B—C—D—E—F—G—H—F H figure 6.2. Matrice d'adjacence du graphe - Quelles sont les

Quelles sont les chaînes les plus courtes entre deux sommets d'un graphe ? Nous avons ainsi constitué, dans le but de disposer d'outils capables de manipuler plusieurs graphes indépendamment les uns des autres au sein d'un même programme, une boîte à outils organisée sous la forme d'une classe d'objets. Nous allons en détailler les principaux méthodes : `■` `Ch` ou `ChC` ou `FM` `Xi` `CTI` `Co`, ou composante-conexes `■` Cette méthode permet de déterminer la composante connexe d'un sommet du graphe. Elle est notamment utilisée pour l'obtention de la liste des composantes du graphe, ainsi que pour tester l'existence d'une chaîne entre deux sommets. recherche chaînes `■` Cette méthode

intériorité si l'on se souvient que le constructeur d'un objet g rap h a p e ne mentionne que la liste d'adjacence, dont la méthode « *deduit* » le nombre d'arêtes. Nous n'en dirons pas plus en ce qui concerne quelques méthodes secondaires, néanmoins fort utiles, ayant trait par exemple à la détermination du nombre de sommets et d'arêtes d'un graphe... dont le lecteur pourra prendre connaissance à la lecture des fichiers disponibles en téléchargement. Les méthodes évoquées ci-dessus peuvent être à leur tour nous n'aurions invoquées pour construire des méthodes fournissant les réponses aux questions posées plus haut. 1.3 Parcours eulériens Rappelons qu'une chaîne ou un cycle eulériens désignent

Dans l'affirmative, on utilise alors la méthode « recherche Euler » pour la génération d'une chaîne ou cycle grappe, `grappe.py`... le début de la classe est pas de deux méthodes : `def grappe` et `def réduits(self, a, b) :`... pour étudier la question de l'existence d'un graminde de Fleury sur lequel nous avons basé notre méthode qui génère un parcours eulérien d'une chaîne ou cycle suivant les grappes. Nous disposons dans notre objet `grappe` de deux méthodes : `def hamiltonien` et `def réduits`... pour étudier la question de l'existence d'un hamiltonien et `def réduits`... pour étudier la question de l'existence d'un hamiltonien.

[illegible]

```
e = Euler's (e) f = Fibonacci de F de Urry .....
g = self (g)
q = self (q)
1) if fin : retu m a s u m m e r y : www.biblio-scientifique.net 242 P r o g r a m m a t i o n P y t h o n p o u r l e s m a t h e m a t i q u e s P = [1] choix = [1] for q in range(1, sommets(s) : if q degre : sommets(s) % 2 == 1 : choix.append(s) if len (choix) == 1 : choix = q, sommets(0) a = choix [ random.randrange(0, len (choix)) - 1] while not
```

```

fin : U = g,a,j,x,c,n,t(s) V = l for s in u; if not gest_u.pont(a,s).v.append(s)b := Random.randrange(0,l)(n,v()) v.append(a) (g,s,g,p,h,e,r,e,d,u,i(t),a,b)a:=b fin (g.nombre_arêtes() == l).append(a).p.append(d).adjacent(s)[a][0] return R le principe de l'algorithme est le suivant, m o y e n n a la vérification préalable de l'existence d'une solution. a) O A) Choisir au hasard u n s o m m e t b) b) Choisir une arête parmi i toutes les arêtes adjacentes (au g r a p h e et de départ) qui ne sont pas un pont, et la conserver dans le parcours en construction. c=3 ->HJD OCM c) vs CTT >= 0 O U R U R Reom me prendre à la seconde étape, en p a r t a n t du nouveau graph

```

graphique de degré impair, lorsqu'il en existe, afin de construire dans ce cas un premier en eulérien. À défaut, un s ou m tel que s et m soient choisis alternativement afin de construire un cycle eulérien. Remarquons qu'au 2^o à la seconde étape, lorsque le seul choix possible est celui d'un p pont, cela signifie que le s ou m et de départ est alors de degré 1, auquel cas le graphe réduit qui en résulte doit avoir au p le 2^o de ce s ou m et isolé, afin de préserver la connexité. www.biblio-science.net C'est la scs 243 Fig. nr 6.3. Graphe non planaire en illustration de test 11 (Terminal %test graphes p-test7 C\U\["F-test8 3\4\["-test92\1\["-test10a\["e\["

quatre opérations et exponentiation) calculant convenablement m et n sur les nom bres rationnels. M'entendions nous les cas d'item en t'existence (depuis la version 2.6 du langage) des nom bres $\frac{a}{b}$ et $\frac{a}{b} \times d$ et $\frac{a}{b} \times c$ im a l. p y), r présentés plus avant, qui am éliorent la gestion des calculs avec ce type de nom bres. Les classes pr ésentées ici so nt bien sûr plus m odestes, mais décrites dans un b u d t étude. Nous utiliserons par ailleurs des instances de classes que nous com poserons soigneusement en t p r u nclure une gestion des erreurs qui p e

scientifique, met 2447 0 0 g r a m m a t i o n e n P y t h o n p o u r l e s m a t h é m a t i q u e s . L e n o m b r e d é c i m a l e s t u n r a t i o n n e l d é n o m i n a t e u r p e u t s'exprimer sous la forme $\frac{a}{b}$, où a et b sont des entiers naturels. Ce choix peut paraître déroulant, mais il est motivé par les raisons suivantes : la conception d'un langage et d'un environnement de programmation se heurte toujours aux limites de temps, à la complexité, et surtout de mémoire de travail. Il en résulte des choix inévitables dont celui qui casse notre conception mathématique moderne des nombres : nous ne pouvons re-présenter qu'un nombre fini de nombres, en nature comme en quantité, dans une mémoire de

[illegible]

opérations et l'expérimentation, ainsi que la possibilité pour le système d'évaluer les expressions ou s'enchaîner les différentes opérations. Nous devons envisager les situations pour lesquelles le système sera placé devant l'évaluation d'un nombre mathématiquement indéfini, afin d'éviter d'un comportement du système dans un tel cas.

x) la notation $\lfloor x \rfloor$ est définie par : $\forall x \in \mathbb{R}, \lfloor x \rfloor = \min (\{n \in \mathbb{Z} \text{ et } n \leq x\})$; la notation $\lceil x \rceil$, que nous appellerons «partie fractionnaire» du réel x , définie par : $\forall x \in \mathbb{R}, \lceil x \rceil = x - \lfloor x \rfloor$.

* Mathématiques concrètes »- Graham, Knuth, Pataschnick. www.biblio-scientifique.net C la sse 245 Avec ces définitions, notions la correspondance suivante entre int (x) et la fonction : $\lfloor x \rfloor$ pour tout $\text{jceI}(x)$, sinon $\lfloor x \rfloor$ La distinction est importante entre la fonction mathématique « partie entière » et le résultat obtenu par transtypage d'un nombre

[illegible]

d'entiers machine dans lesquels les entiers machine sont considérés comme valides. Ainsi, chaque opération arithmétique doit contrôler elle-même si son résultat mathématique appartient effectivement ou non à l'intervalle de validité défini. b) Le langage Python offre un module additionnel decimal pour la gestion fine des nombres décimaux, avec une précision de calcul bien supérieure à celle offerte par le type float à standard.

[illegible]

mais inutilisable! Sans entrer ici dans le détail, disons simplement que l'erreur provient de la représentation interne des nombres en P y th o n (ce défaut existe dans la majorité des langages, pour des raisons de conception des unités logiques de calcul). Nous reviendrons sur ces questions dans un autre chapitre. 2.3 Une seconde tentative Nous

[illegible]

