

**Objectifs: maitriser les capacités suivantes:**

Entrées/sorties, affectations : Calcul de l'IMC .....	2
Instruction conditionnelle Si ...Alors ...[sinon...] : Résolution d'équation....	4
Instruction conditionnelle Si ...Alors[sinon...] Triangles rectangles .....	5
Première recherche d'un maximum Triangles rectangles .....	6
Introduction des boucles "Répéter Jusqu'à" et "Tant que" calcul d'une somme : Passage à la caisse 1.....	7
Introduction d'un compteur Passage à la caisse 2.....	8
Deuxième recherche d'un maximum : Passage à la caisse 3.....	9
Utiliser les boucles "Pour" : afficher les carrés de nombres entiers.....	10
Utiliser les boucles "Pour" : une somme curieuse.....	11
Imbriquer des boucles : table de multiplication.....	12
Générer des nombres aléatoires .....	13
Stocker dans une liste .....	14
Programmer des instructions graphiques.....	15

[Consulter le tableau de conversion des langages](#)

**Notes:** l'algorithmique ne devrait pas constituer un chapitre à part dans le cours de seconde. Chaque compétence peut être introduite lorsque son usage se présente dans la progression de l'année.

De nombreuses applications de ces capacités sont illustrées dans le document ressources : [algorithmique pour la classe de seconde](#)

**Note sur la rédaction:** dans les tableaux ci-dessous : les instructions simples peuvent être remplacées par des blocs d'instructions, en utilisant la syntaxe adéquate, voir le paragraphe:

Instruction conditionnelle Si ...Alors ...[sinon...] Résolution d'équation

**ATTENTION :** pour les calculatrices courantes, les mots du langage ne doivent pas être tapés lettre par lettre, on les trouve sur les touches, dans le catalogue ou dans les menus. Ci-dessous ils sont en caractère gras.



## Entrée/sorties, affectations

## Calcul de l'IMC

Langage algorithmique	Scilab	Python 2.6	TI	Casio	XCAS
<u>Saisir a</u>	<code>a=input ( "donner a ")</code>	<code>a=input('donner a ')</code> pour traiter un calcul sur des entiers, ou sur des décimaux : <code>a=int(input('donner a '))</code> ou <code>a=float(input('donner a '))</code>	<b>Prompt A</b> ou <b>Input "A=", A</b>	"A":.?→A	<code>input("a= ",a);</code> ou <code>saisir ("a= ",a) ;</code>
<u>Afficher a</u>	<code>a</code> ou <code>afficher( " a=" +string(a))</code> ou <code>disp( " a=" +string(a))</code>	<code>print 'a= ', a</code>	<b>Disp "A=", A</b>	"A=" : A▲	<code>print ("a=" + a);</code> ou <code>afficher("a="+a);</code>
affectation: <u>b prend la valeur a,</u>	<code>b = a</code>	<code>b = a</code>	A <b>STO</b> B	<code>a → b</code>	<code>b = a ;</code>

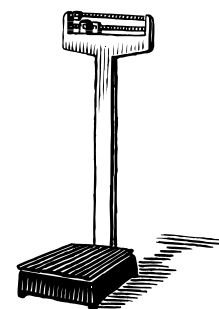


**Énoncé 1:** L'I.M.C.<sup>(\*)</sup> se calcule en divisant la masse en kg d'un individu par le carré de sa taille exprimée en mètre.

- Ecrire un algorithme qui permet de calculer et d'afficher un I.M.C.

Préciser les entrées et les sorties, le nom et la signification des variables

- Programmer l'algorithme sur calculatrice ou ordinateur.
- Tester ce programme de façon à détecter des erreurs éventuelles.



- L'I.M.C. de l'ogre : un ogre de 3 m a une masse de 410 kg,
  - sur calculatrice : faite afficher son I.M.C. sous forme de fraction.
  - Sur ordinateur, faire afficher son I.M.C. sous forme décimale approchée avec deux chiffres après la virgule

[Consulter le tableau de conversion des langages](#)

### Capacités visées

Pour le professeur

- Identifier les variables, celles qui sont en entrée, celles qui sont en sortie
- utiliser une affectation
- gérer des entrées sorties (instructions Saisir, Afficher)
- prendre en main un environnement de programmation sur calculatrice ou ordinateur
- découvrir progressivement la représentation des nombres dans la machine, maîtriser l'affichage d'un résultat

[Retour au sommaire](#)

[Corrigés](#) [Algorithmes](#) [Scilab](#) [Python](#) [Xcas](#) [TI](#) [CASIO](#)

(\*) Si le calcul de l' I.M.C.(Indice de masse corporelle) peut engendrer dans certaines classes des comparaisons à éviter, on peut le remplacer par n'importe quelle grandeur qui est fonction de plusieurs variables, et telle que le traitement par la calculatrice ne soit pas possible directement. (les calculatrices courantes ne traitent que les fonctions à une variable). Voir des exemples dans les prolongements ci-dessous.



## Prolongements, par exemple pour un en travail à la maison

Cercles, disques et boules

Entrée/sorties, affectations

### Énoncé 2

Ecrire un algorithme qui, à partir d'un rayon  $r$ , calcule la longueur du cercle de rayon  $r$ , l'aire du disque de rayon  $r$ , le volume de la boule de rayon  $r$ . Préciser les entrées et les sorties

Programmer l'algorithme sur calculatrice ou ordinateur :

Comment peut-on tester le programme ?



*Cet exercice n'est pas corrigé*

### Et aussi:

- Algorithmes de calcul des aires usuelles (triangle, trapèze), ou des volumes usuels (prismes, pyramides et cônes) au programme de collège.
- Algorithme qui donne l'équation réduite de droite non parallèle à l'axe des ordonnées

[Retour au sommaire](#)

## Instruction conditionnelle Si ..Alors ...[sinon...] Résolution d'équation

	Scilab	Python 2.6	TI	Casio	XCAS
<p style="color: blue;">Si <i>condition</i> Alors <i>Instruction1</i> <i>facultatif*</i> [ <i>Sinon</i> <i>Instruction2</i> ] Fin du si</p>	<pre style="color: blue;">if <i>condition</i> then   <i>Instructions1</i> else   <i>Intructions2</i> End</pre> <p style="color: green; font-size: small;">(if et then doivent être sur la même ligne)</p>	<pre style="color: blue;">if <i>condition</i> :     <i>Instruction1</i> [ else :     <i>Intruction2</i> ]</pre>	<pre style="color: blue;">If <i>condition</i> Then <i>Instruction1</i> [Else <i>Intruction2</i>] End</pre>	<pre style="color: blue;">If <i>condition</i> Then <i>Instruction1</i> [Else <i>Intruction</i>)2] IfEnd</pre>	<pre style="color: blue;">if ( <i>condition</i> )   {<i>Instruction1</i> } [ else   {<i>Intruction2</i> }]; ou en français si <i>condition</i> alors <i>instruction1</i> [<i>sinon</i> <i>instruction2</i>] fsi</pre>
<p style="color: blue;">Tests =, ≠, ≤, ≥</p>	<p style="color: blue;">==, &lt;&gt;, &lt;=, &gt;=</p>	<p style="color: blue;">==, !=, &lt;=, &gt;=</p>	<p style="color: blue;">=, ≠, ≤, ≥</p>	<p style="color: blue;">=, ≠, ≤, ≥</p>	<p style="color: blue;">=, !=, &lt;=, &gt;=</p>
<p style="color: blue;">Bloc d'instructions</p>	<p style="color: blue;">Le bloc est déterminé par la structure de l'instruction</p>	<p style="color: blue;">Les instructions d'un bloc ont la même marge à gauche</p>	<p style="color: blue;">Le bloc est terminé par <b>End</b></p>	<p style="color: blue;">Le bloc est terminé par <b>End, ifEnd, whileEnd, ...</b></p>	<p style="color: blue;">En anglais, Le bloc est encadré par des accolades: { <i>instruction1</i> <i>Instruction2</i> etc }</p>

\* si l'instruction comporte "sinon", ne pas écrire les crochets !

### Énoncé 3

On se propose d'écrire un programme qui résout automatiquement les équations du type  $ax+b=c$ , où  $a$ ,  $b$  et  $c$  sont trois nombres réels donnés, et  $x$  est l'inconnue réelle.

1) Résoudre dans l'ensemble des nombres réels les équations d'inconnue  $x$ :

a)  $3x+5=-1$

b)  $x\sqrt{2}-4=1$

c)  $0x+3=7$

d)  $0x+1=1$

2) Dans le cas général, déterminer la valeur de  $x$  en fonction des nombres  $a$ ,  $b$  et  $c$ .

3) Ecrire un algorithme qui, à partir de la donnée des trois nombres  $a$ ,  $b$  et  $c$ , fournit la résolution de l'équation  $ax+b=c$ .

4) Programmer l'algorithme sur calculatrice ou ordinateur, puis expliquer comment tester le programme.

[Consulter le tableau de conversion des langages](#)

Cet exercice n'est pas corrigé

**Et aussi:**

Algorithme qui donne l'équation réduite d'une droite définie par deux points distincts (test du cas où la droite est parallèle à l'axe des ordonnées)

Cet algorithme peut être abordé en début d'année avec la géométrie repérée, ou avec l'étude des fonctions.

Corrigé : voir atelier Algorithmique des journées d'animation 2009 [consulter](#)

[Retour au sommaire](#)

## Instruction conditionnelle Si ...Alors[sinon...] Triangles rectangles

### Énoncé 4.

- a) Ecrire un algorithme qui, à partir de trois longueurs  $a$ ,  $b$  et  $c$ , détermine si le triangle correspondant est rectangle. Les trois côtés étant fournis dans l'ordre croissant. Préciser les entrées, les sorties. Programmer l'algorithme sur calculatrice ou ordinateur :  
Quels nombres simples utilisez-vous pour tester le programme ?
- b) Un triangle qui a pour dimensions, 5, 12 et 13 est-il rectangle?



Même question avec un triangle qui a pour dimensions  $\frac{5}{3}$ ,  $\frac{12}{3}$ , et  $\frac{13}{3}$ .

Puis avec 3, 4, et  $5+10^{-12}$ .

( Pour répondre, tester ce triplet avec le programme et sans le programme.

Expliquer ce que l'on observe sur calculatrice (non formelle).

Effectuer des tests avec d'autres types de nombres, par exemple des irrationnels).

- c) Soit  $u$  et  $v$  deux réels strictement positifs tels que  $u < v$ .  
On pose:  $a = v^2 - u^2$ ,  $b = 2uv$  et  $c = u^2 + v^2$ .  
Le triangle qui pour dimensions  $a$ ,  $b$  et  $c$  est-il rectangle?
- d) Trouver trois entiers naturels autres que 3, 4 et 5, et autres que 5, 12 et 13, tels que le triangle de dimensions  $a$ ,  $b$  et  $c$  soit rectangle.
- e) Modifier l'algorithme et le programme de façon à afficher en plus : « données incorrectes », si jamais  $c$  n'est pas le plus grand des trois nombres rentrés, et sinon, afficher si le triangle est rectangle ou non.

[Consulter le tableau de conversion des langages](#)

[Retour au sommaire](#)

[Corrigés](#) [Algorithmes](#) [Scilab](#) [Python](#) [Xcas](#) [TI](#) [CASIO](#)

## Première recherche d'un maximum      Triangles rectangles

### Énoncé 4.bis

Ecrire un algorithme qui, à partir de trois longueurs  $a$ ,  $b$  et  $c$ , détermine quelle est la plus grande de ces longueurs. Puis l'algorithme détermine si le triangle correspondant est rectangle ou non. Les trois côtés ne sont pas forcément fournis dans l'ordre croissant.

[Consulter le tableau de conversion des langages](#)

Capacités visées en plus des précédentes

*Pour le professeur*

- utiliser une instruction conditionnelle "Si ... Alors... Fin du Si"
- distinguer une égalité d'une affectation
- observer les problèmes de la gestion des égalités et des valeurs approchées dans la machine

- dans l'énoncé 4.b: on recherche un maximum. Pour que l'élève ait bien le temps de le chercher, ce deuxième énoncé peut être donné à la maison. La question concernant le tri des trois nombres en ordre croissant est un exercice de réflexion purement algorithmique : on n'effectue aucun calcul dans cette partie. Bien sûr, si l'on met les 3 nombres dans une liste, les fonctions d'une calculatrice assez basique permettent d'en obtenir immédiatement le maximum, le minimum ou même de les classer.

Compétences mathématiques:

- réviser la réciproque du théorème de Pythagore
- Tester des triplets de Pythagore génériques : travail sur le calcul littéral et les identités remarquables.

**Prolongements:** A partir des résultats de la question 3, on peut tester le programme avec des grands nombres et mettre en évidence les limites d'une calculatrice, ainsi que les problèmes d'arrondis.

[Retour au sommaire](#)

[Corrigés](#)   [Algorithmes](#)   [Scilab](#)   [Python](#)   [Xcas](#)   [TI](#)   [CASIO](#)





## Introduction d'un compteur : Passage à la caisse 2

**Énoncé 5.b** Compléter l'algorithme précédent en introduisant une nouvelle variable qui compte les articles. Compléter les informations fournies en donnant le nombre d'articles et le prix moyen des articles achetés.

[Consulter le tableau de conversion des langages](#)

[Corrigés](#) [Algorithmes](#) [Scilab](#) [Python](#) [Xcas](#) [TI](#) [CASIO](#)

[Retour au sommaire](#)





## Deuxième recherche d'un maximum : Passage à la caisse 3

**Énoncé 5.c** Compléter l'algorithme précédent de façon à repérer le montant de l'article le plus cher. Afficher aussi le prix de cet article.

Lors d'une promotion, le magasin accorde une remise de 15% sur un article (le plus cher, évidemment). Faire afficher la réduction, et le nouveau total à payer.

[Consulter le tableau de conversion des langages](#)

Capacités visées :

Pour le professeur

- Être capable de traduire la répétition d'une instruction.

Le type de boucle le plus intuitif et le plus simple à mettre en œuvre est :

"Répéter... Jusqu'à..."

(Voir le document: [Choisir le type de boucle](#))

- comprendre une instruction du type S prend la valeur  $S+x$ , être capable de calculer une somme.

- Être capable de gérer un compteur : cas particulier de somme :  $x$  prend la valeur  $x+1$

On réinvestit aussi ici la recherche d'un maximum.

Compétences mathématiques : augmentations, réductions, et pourcentages.

*Cet exercice permet de comprendre certaines fonctionnalités portant sur les listes : on peut obtenir aussi ces résultats par l'utilisation des fonctions sur les listes ( sur calculatrice : menu LIST : on met tous les prix dans une même liste  $i$ , puis on demande :  $\text{Sum}(\text{List } i)$ ,  $\text{max}(\text{List } i)$ , ensuite  $0,15 \times \text{max}(\text{List } i)$  et enfin  $\text{SUM List } i - 0,15 \times \text{max}(\text{List } i)$*

*On pourrait le traiter de la même manière sur un tableur.*

*Les différences avec l'écriture d'un programme :*

- Le programmeur a la liberté de choisir plus précisément ce qu'il veut obtenir, avec éventuellement la satisfaction de créer.

- Avec l'utilisation des listes, la gestion de la mémoire est loin d'être optimisée : dans l'algorithme, trois mémoires suffisent, même si l'on rentre 1000 prix !

[Retour au sommaire](#)

## Utiliser les boucles "Pour" afficher les carrés de nombres entiers

Exemples:

	Scilab	Python 2.6	TI	Casio	XCAS
Pour i variant de 1 à n Faire Instruction(s) Fin du pour	<pre>for k=1:n   Instruction(s) end</pre>	<pre>for i in range(1,n+1):   Instruction(s)</pre>	<pre>For (I,1,N)   Instruction(s) End</pre>	<pre>For 1→I To N   Instruction(s) Next</pre>	<pre>for (k :=1 ;k&lt;=n ;k :=k+1)   {Instructions};   ou en français pour k de 1 jusque n faire   Instruction(s); fpour (ATENTION : lettre i minuscule à proscrire, c'est un nombre complexe)</pre>

Dans ces boucles, le nombre entier  $i$  prend successivement les valeurs  $1, 2, \dots, n$ , où  $n$  désigne un entier supérieur ou égal à 1.

### Énoncé 6

- 1) Ecrire un algorithme qui affiche les carrés des nombres entiers compris entre 1 et 20.
- 2) Programmer l'algorithme sur calculatrice ou ordinateur.
- 3) Parmi les nombres suivants, dire ceux qui sont le carré d'un nombre entier, sans utiliser la touche racine carrée de la calculatrice:  
44, 64, 61, 81, 96, 125, 121, 144, 156, 169, 196, 200, 225, 250, 256, 264, 289, 300, 400.

[Consulter le tableau de conversion des langages](#)

[Retour au sommaire](#)

[Corrigés](#) [Algorithmes](#) [Scilab](#) [Python](#) [Xcas](#) [TI](#) [CASIO](#)

### Et aussi, pour appliquer des boucles pour

- Calculer d'autres sommes (voir l'exercice suivant, par exemple)
- Faire calculer un tableau de valeurs de fonction
- Construire un polygone régulier à  $n$  sommets
- Programmer  $n$  sauts aléatoires d'une puce sur un axe
- Rechercher les diviseurs d'un entier naturel
- Etc.

## Utiliser les boucles "Pour" : Une somme curieuse



### Énoncé 7

On a vu dans l'exercice 5 comment calculer une somme. On va maintenant étudier des sommes particulières : les sommes des premiers nombres impairs.

Par exemple: la somme des trois premiers nombres impairs est :  $1+3+5$ .

On peut remarquer que tout nombre impair est un nombre de la forme  $2p+1$ , où  $p$  désigne un entier naturel. Par exemple 17 peut s'écrire  $2 \times 8 + 1$ , autrement dit le nombre 17 correspond au nombre  $p$  égal à 8.

1)a) Vérifier que  $1+3+5$  est la somme des nombres impairs correspondant à  $p=0$ ,  $p=1$ , et  $p=2$ .

b) Quelle est la somme des nombres impairs obtenue si  $p$  prend les valeurs  $0,1,2, \dots,5$ .

c) trouver l'entier naturel  $n$  tel que si  $p$  prend les valeurs  $0,1,2, \dots,n$ , alors la somme des nombre impairs correspondants vaut 441.

2) Ecrire un algorithme qui permet de saisir un entier naturel  $n$ , puis de calculer la somme des nombres impairs correspondants aux valeurs  $0,1,2, \dots,n$  du nombre  $p$  défini ci-dessus. Programmer l'algorithme sur calculatrice ou ordinateur, tester le programme.

3) En observant les résultats obtenus, proposer une formule qui permet de calculer beaucoup plus rapidement cette somme.

4) Calculer la somme  $S=1+3+5+\dots+200\,001$ , correspondant aux valeurs de  $p$ :  $0,1, 2, \dots, 100\,000$ .

[Consulter le tableau de conversion des langages](#)

*Cet exercice n'est pas corrigé*

### Notes:

*Pour le professeur*

1- Cette vision des nombres impairs peut occasionner quelques difficultés de compréhension en classe de seconde, on peut déjà commencer par l'écriture des nombres pairs sous la forme  $2p$ , puis faire remarquer que tout nombre impair suit un nombre pair.

2- A la question 4), le nombre 100 000 est choisi pour forcer l'utilisation de la formule trouvée à la question 3). En effet sur les calculatrices courantes le programme met plusieurs dizaines de minutes à s'exécuter et fournit un résultat approché sous forme scientifique.

[Retour au sommaire](#)

## Imbriquer des boucles      table de multiplication

### Énoncé 8

Programmer l'écriture d'une table de multiplication des entiers naturels inférieurs ou égaux à 10.

[Consulter le tableau de conversion des langages](#)

Notes:

*Pour le professeur*

L'utilisation de boucles imbriquées revient dans de nombreux algorithmes, cet exemple ne présente qu'une première approche, pour montrer qu'on a le droit...

[Corrigés](#)   [Algorithmes](#)   [Scilab](#)   [Python](#)   [Xcas](#)   [TI](#)   [CASIO](#)

[Retour au sommaire](#)

## Générer des nombres aléatoires

	<b>Scilab</b>	<b>Python 2.6</b>	<b>TI</b>	<b>Casio</b>	<b>XCAS</b>
Nombres aléatoires	<b>Module Lycée</b>	<b>Avec from random import*</b>	<b>Touche MATH/ PRB</b>	<b>Run / touche OPTN / PROB</b>	<b>Le point virgule sépare les instructions</b>
Initialisation		seed()			srand
Nombre réel aléatoire dans [ 0;1[	tirage_reel (1,0,1) <i>(liste de 1 seul réel aléatoire)</i>	random()	<b>rand</b>	<b>Rand#</b>	rand(0,1) ou alea(0,1)
Nombre réel aléatoire dans [ a;b[	tirage_reel(p,a,b) <i>(liste de p réels aléatoires)</i>	a + (b-a) x random()	a + (b-a) x <b>rand</b>	a + (b-a) x <b>Rand#</b>	rand(a,b) ou alea(a,b)
Entier aléatoire dans {a ;a+1 ; ... ; b} avec a et b entiers	tirage_entier(p,a,b) <i>(liste de p nombres entiers aléatoires)</i>	randint(a,b)	<b>rand(a,b)</b>	a + <b>Intg</b> ((b-a+1)x <b>Rand#</b> )	a+rand(b-a+1) ou a+alea(b-a+1)
Exemple : entier aléatoire dans : {0 ;1} puis dans {1 ;2 ;3 ;4 ;5 ;6}	L= tirage_entier (1,0,1) <i>(liste de 1 seul entier aléatoire)</i> L= tirage_entier (1,1,6)	randint(0,1) randint(1,6)	<b>rand(0,1)</b> <b>randInt(1,6)</b>	<b>Intg(2*Rand#)</b> <b>1+ Intg(6*Rand#)</b>	rand(2) ou alea(2) 1+rand(6) ou 1+alea(6)

- Pour consulter des activités sur la génération de nombres aléatoires appliquée aux probabilités : consulter les activités spécifiques de la fiche thématique Algorithmique : Algorithmique et probabilités [consulter](#)

[Retour au sommaire](#)

## Stocker dans une liste

	Scilab	Python 2.6	TI	Casio	XCAS
Créer une liste	<i>En Scilab, les listes sont aussi appelées vecteurs</i> $l=[5,8,9]$ $l(1)$ vaut 5, $l(2)$ vaut 8...	$l=[5,8,9]$ l'indice commence à <u>zéro</u> , $l[0]$ vaut 5, $l[1]$ vaut 8,...	Les listes $L_1, L_2$ , existent dans le mode Statistique	Les listes <b>List 1, List 2</b> existent dans le menu STAT	$l:=[5,8,9]$ l'indice commence à <u>zéro</u> , $l[0]$ vaut 5, $l[1]$ vaut 8,...
Vider une liste $l$ Créer une liste vide (python, Xcas)		$l=[]$	<b>ClrList</b>	Menu <i>Stat</i> puis <b>DEL-A</b> ou $\{0\} \rightarrow$ List 1	$l := []$
Créer et remplir une liste de six 0 Créer $l$ $=[5^2,7^2,9^2,11^2,13^2]$	$l = \text{zeros}(1,10)$ <i>Avec une boucle, ou...</i>	<i>Avec une boucle et la fonction : append</i>	<b>6 STO dim(L<sub>1</sub>)</b> Seq(X <sup>2</sup> ,X,5,13,2) <b>STO</b> L <sub>1</sub>	$6 \rightarrow$ Dim List 1 Dans OPTN LIST Seq(X <sup>2</sup> ,X,5,13,2) $\rightarrow$ List 1	$l := [0\$6]$ ou $l := [0\$(k=1..6)]$ $l := \text{seq}(k^2, k=5..13, 2);$
Ajouter un élément $a$ à la fin de la liste $l$	<i>Si <math>l</math> comporte déjà <math>n</math> éléments : <math>l(n+1)=a</math></i>	$l.append(a)$	$a$ <b>STO</b> L <sub>1</sub> (I) I étant le premier indice non encore utilisé	Dans le menu <i>List</i> , entrer directement l'élément $a$ à la fin de la liste ! (inutilisable dans un programme)	$l := \text{append}(l, a)$
Accès à l'élément numéro $k$	$l(k)$	$l[k]$	L <sub>1</sub> (k)	List1[k]	$l[k]$
Longueur d'une liste	$\text{taille}(l)$	$\text{len}(l)$	<b>dim</b> (L <sub>1</sub> )	Dim List 1 Dans OPTN LIST	$\text{length}(l)$
Remplir une liste $l$ avec dix nombres Aléatoires pris dans {1 ;2 ;3 ;4 ;5 ;6}	$l = \text{tirage\_entier}(10,1,6)$	<i>Avec une boucle et append</i>	Seq( <b>randInt</b> (1,6),X,1,10,1)	Seq(1+ <b>Intg</b> (6* <b>Rand</b> #),X,1,10,1) Seq : Dans OPTN LIST	$l := [(1+\text{rand}(6))\$(k=1..10)]$
Remplir une liste Avec dix nombres Aléatoires pris dans {1 ;2 ;3 ;4 ;5 ;6}	$l = \text{tirage\_entier}(10,1,6)$	<i>Avec une boucle et append</i>	Seq( <b>randInt</b> (1,6),X,1,10,1)	Seq(1+ <b>nt</b> (6* <b>Rand</b> #),X,1,10,1)	$l := [(1+\text{rand}(6))\$(k=1..10)];$

➤ Pour consulter des activités sur la génération de nombres aléatoires appliquée aux probabilités : consulter les activités spécifiques de la fiche thématique Algorithmique : Algorithmique et probabilités [consulter](#)

[Retour au sommaire](#)

## Programmer des instructions graphiques

Quelques instructions pour les graphiques					
	Scilab	Python 2.6 uniquement <i>from graphseconde import*</i> télécharger module et notice	TI 82-84	Casio 35+	XCAS 0.8.4
Passer en mode graphique / mode calcul	<i>automatique</i>	<i>fenetre(xmin,xmax,ymin,ymax)</i> <i>axes()</i>	<i>DispGraph</i>	<i>DrawGraph</i> <small>Dans shift PRGM DISP Grph</small>	<i>DispG</i> <i>DispHome</i>
Effacer l'écran Graphique	<i>Clf</i> <b>Attention : clear efface les fonctions !</b>		<i>ClrDraw</i> ou <i>EffDessin</i>	<i>ClrGraph</i> <small>Dans shift PRGM CLR</small>	<i>ClrGraph</i> <i>efface</i>
Placer un point M(x,y)	<i>plot(x,y, ". ")</i>	<i>point(x,y[,couleur])</i>	<i>Pt-On(x,y[,marque])</i>	<i>Plot x,y</i> <small>Dans shift Sketch(F4)</small>	<i>point(x,y)</i>
Tracer le segment [AB] avec A(x <sub>A</sub> ; y <sub>A</sub> ) et B(x <sub>B</sub> ; y <sub>B</sub> )	<i>plot([x<sub>A</sub>, x<sub>B</sub>], [y<sub>A</sub>, y<sub>B</sub>])</i> <b>Attention à l'ordre!</b>	<i>segment((x<sub>A</sub>, y<sub>A</sub>, x<sub>B</sub>, y<sub>B</sub> [,couleur])</i>	<i>Line(x<sub>A</sub>, y<sub>A</sub>, x<sub>B</sub>, y<sub>B</sub>)</i> ou <i>Ligne(x<sub>A</sub>, y<sub>A</sub>, x<sub>B</sub>, y<sub>B</sub>)</i>	<i>F-line x<sub>A</sub>, y<sub>A</sub>, x<sub>B</sub>, y<sub>B</sub></i> <small>Dans shift Sketch(F4)</small>	<i>A :=point(x<sub>A</sub>; y<sub>A</sub>);</i> <i>B:=point(x<sub>B</sub>; y<sub>B</sub>);</i> <i>segment(A,B);</i>
Tracer un cercle	<i>t=linspace(0,2*pi,100);</i> <i>x=x<sub>0</sub>+r*cos(t);y=y<sub>0</sub>+r*sin(t);</i> <i>plot(x,y)</i>	<i>cercle_cr(x,y,r[,couleur])</i> <i>cercle_cp(x,y,s,t[,couleur])</i>	<i>Circle(x,y,r)</i> <i>x et y coordonnées du centre et r le rayon</i>	<i>Circle x,y,r</i> <i>x et y coordonnées du centre, et r le rayon</i> <small>Dans shift Sketch(F4)</small>	<i>circle(point(x,y),r)</i> <i>voir autres possibilités dans l'index</i>

➤ Sur la Casio GRAPH 25 :

ClrGraph : SHIFT Prgm ▶ CLR Grph

Plot : SHIFT Sketch PLOT Plot

F-LINE: SHIFT Sketch LINE F-LINE

➤ des activités sur la construction de graphiques: consulter les activités spécifiques de la fiche thématique Algorithmique : [consulter](#)

[Retour au sommaire](#)

## Consulter le tableau complet de conversion des langages