

```

s = experience()
n = 1
L = [s] # moyenne de
while n < nExperiences:
    n = n+1
    s = s + experience()
    L.append(s/n) # o
plt.plot(list(range(1,
plt.plot([1, nExperiences

```

FICHE n°4 : PROGRAMMER L'INSTRUCTION CONDITIONNELLE

Syntaxe :

Langage naturel	TI	CASIO
Si Condition	:If condition	If condition↵
Alors Instructions1	:Then	Then Instruction1↵
Sinon	:Instruction1	Else Instruction2↵
Instructions2	:Else	IfEnd↵
Fin Si	:Instruction2	
	:End	

Exercice 1 :

- Expliquer le principe de l'algorithme ci-contre.
 - Ce même algorithme peut se traduire par le programme ci-dessous.
- Quelle valeur obtient-on pour B et C lorsqu'on saisit A = 182 au départ ?
Qu'affiche l'algorithme en sortie dans ce cas.

```

Saisir A
Affecter à B la valeur A/13
Affecter à C la valeur arrondie à l'unité de B
Si B = C
Alors afficher "A est divisible par 13"
Sinon
Afficher "A n'est pas divisible par 13"
Fin Si

```

TI	CASIO
<pre> PROGRAM:DIVISIBI :Input A :A/13→B :round(B,0)→C :If B=C :Then :Disp "EST DIV PAR 13" :Else :Disp "N EST PAS DIV PAR 13" :End </pre>	<pre> =====DIVISIBI===== "A="?→A↵ A÷13→B↵ RndFix(B,0)→C↵ If B=C↵ Then "EST DIV PAR 13", Else "N EST PAS DIV PAR IfEnd↵ </pre>

- Modifier le programme dans le but de vérifier si un nombre est divisible par 29.
 - Les nombres suivants sont-ils divisibles par 29 ?
565 – 6785 – 646 195 034 – 1 970 659 794

Exercice 2 :

Ecrire un programme permettant de vérifier si un nombre donné est divisible par 13 en effectuant un test sur le reste de la division de ce nombre par 13.

Langage naturel	TI	CASIO
Reste de la division euclidienne de A par B	A–B*ent(A/B)	A–BxInt (A÷B)



Hors du cadre de la classe, aucune reproduction, même partielle, autres que celles prévues à l'article L 122-5 du code de la propriété intellectuelle, ne peut être faite de ce site sans l'autorisation expresse de l'auteur.
www.maths-et-tiques.fr/index.php/mentions-legales

Exercice 3 :

Dans le programme ci-dessous traduisant l'algorithme ci-contre, les instructions conditionnelles ont été supprimées.

TI	CASIO
<pre> Input X Input Y X<5→Y 10×X→X 10+Y→Y Disp X*Y </pre>	<pre> "X="?→X↵ "Y="?→Y↵ X<5×Y↵ 10×X→X↵ 10×Y→Y↵ X×Y, </pre>

```

Saisir x
Saisir y
Si x < 5y
Alors affecter à x la valeur 10x
Sinon
Affecter à y la valeur 10y
Fin Si
Afficher xy

```

- Corriger en complétant le programme par les instructions conditionnelles manquantes.
- Tester ce programme pour x = 5 et y = 9. Même question pour x = 12 et y = 2.

Exercice 4 :

- Ecrire un programme traduisant l'algorithme ci-contre.
- Tester ce programme pour trouver quelques triplets de Pythagore.

```

Saisir dans l'ordre croissant trois nombres entiers A, B, C
Affecter à M la valeur de A²
Affecter à N la valeur de B²
Affecter à X la valeur de M + N
Affecter à Y la valeur de C²
Si X = Y
Alors afficher "A, B, C est un triplet de Pythagore"
Sinon afficher "A, B, C n'est pas un triplet de Pythagore"
Fin Si

```

Exercice 5 :

Dans le programme ci-contre, les affichages en sortie de l'algorithme ont été supprimés.

- Quel problème permet de résoudre cet algorithme ?
- Compléter le programme par les affichages en sortie manquants.
- Tester ce programme pour différentes valeurs de A et B.

TI	CASIO
<pre> PROGRAM:SIGNE :Input A :Input B :If A>0 :Then :If B>0 :Then :Disp "LE PRODUIT AB" :Disp "EST ..." :Else :Disp "LE PRODUIT AB" :Disp "EST ..." :End :Else :If B>0 :Then :Disp "LE PRODUIT AB" :Disp "EST ..." :Else :Disp "LE PRODUIT AB" :Disp "EST ..." :End :End </pre>	<pre> =====SIGNE ===== "A="?→A↵ "B="?→B↵ If A>0↵ Then If B>0↵ Then "LE PRODUIT AB EST Else "LE PRODUIT AB EST IfEnd↵ Else If B>0↵ Then "LE PRODUIT AB EST Else "LE PRODUIT AB EST IfEnd↵ </pre>

Exercice 6 :

Écrire et tester un programme qui demande en entrée à un client le montant total de ses achats.
En fonction de la somme dépensée, le programme affiche en sortie le prix à payer :
- Si la somme dépensée est strictement inférieure à 75 €, il obtient 5 % de remise.
- Si la somme dépensée est supérieure à 75 €, il obtient 8 % de remise.