

Algorithmes sur calculatrice



1. Résumé des commandes selon la marque

créer, modifier, exécuter

	TI	Casio
accéder au menu	PRGM	MENU PRGM
créer un programme	NEW	NEW
modifier/éditer un programme	EDIT	EDIT
quitter le mode d'édition	QUIT 2nd + MODE	EXIT (autant que nécessaire)
exécuter un programme	EXEC puis ENTER	EXE

mots-clés et caractères particuliers

	TI	Casio
→	STO →	→
Prompt Input Disp	PRGM E/S	SHIFT + PRGM VAR
If Then Else For While End	PRGM	SHIFT + PRGM VAR COM
espace " lettres	ALPHA	ALPHA
= ≠ > ≥ < ≤	TEST 2nd + MATH	SHIFT + PRGM VAR REL

2. Éléments particuliers nécessaires pour la suite

a. Nombre entier aléatoire

On peut obtenir un nombre aléatoire entre deux entiers avec la calculatrice. Par exemple, si l'on souhaite simuler un lancer de dé, donc obtenir un nombre aléatoire entre 1 et 6, on exécute l'instruction suivante :

- En TI : entAleat(1,6)
- En Casio : RanInt \sharp (1,6)

Les fonctions utilisées se trouvent :

- Pour les TI en appuyant sur **MATH** puis en allant sur **PRB**.
- Pour les Casio en appuyant sur **OPTN** puis en allant sur **PROB**.

b. Logique

Le « et » (ou « and »), ainsi que d'autres éléments de logique, s'obtiennent :

- en TI, dans **TEST** **2nd**+**MATH** puis **LOGIC**
- en Casio, dans **OPTN** puis **LOGIC**

c. Caractères particuliers

Certains caractères très particuliers, comme l'apostrophe et le point d'exclamation s'obtiennent :

- Pour les TI, à partir de **CATALOG** **2nd**+**0** , puis **3**.
- Pour les Casio, en mode d'édition de programme, à partir de **CHAR** (utiliser **EXIT** si nécessaire avant).

3. Le lièvre et la tortue

Une partie du jeu du lièvre et de la tortue se déroule ainsi : Le lièvre et la tortue sont tous les deux sur la ligne de départ de la course. La tortue a besoin de 5 déplacements pour atteindre l'arrivée. Le lièvre, beaucoup plus rapide, n'a besoin que d'un déplacement pour atteindre l'arrivée.

Tant qu'aucun des deux n'a atteint l'arrivée, on lance un dé :

- Si le dé tombe sur 1, 2, 3, 4 ou 5, alors la tortue fait un déplacement, le lièvre attend ;
- Si le dé tombe sur 6, alors le lièvre fait son déplacement (et atteint l'arrivée).

Le gagnant est bien sûr celui de la tortue ou du lièvre qui atteint l'arrivée en premier.

1. Qui, à votre avis, gagne le plus souvent : la tortue ou le lièvre ?
2. Entrer l'algorithme suivant dans la calculatrice, en le nommant COURSE :

Langage pseudo-algorithmique :	TI :	Casio :
L prend la valeur 0	0→L	0→L↵
Pour I allant de 1 à 5 Faire	For(I,1,5)	For 1→I To 5↵
Si L=0 Alors	If L=0	If L=0↵
X prend la valeur alea_entier(1,6)	Then	Then RanInt#(1,6)→X↵
Si X=6 Alors	entAleat(1,6)→X	If X=6↵
L prend la valeur 1	If X=6	Then 1→L↵
Sinon	Then	Else "LA TORTUE AVANCE"↵
Afficher "La tortue avance"	1→L	IfEnd↵
FinSi	Else	IfEnd↵
FinSi	Disp "LA TORTUE AVANCE"	Next↵
FinPour	End	If L=1↵
Si L=1 Alors	End	Then "LE LIEVRE A GAGNE"↵
Afficher "Le lièvre gagne"	End	Else "LA TORTUE A GAGNE"↵
Sinon	If L=1	IfEnd
Afficher "La tortue gagne"	Then	
FinSi	Disp "LE LIEVRE GAGNE"	
	Else	
	Disp "LA TORTUE GAGNE"	
	End	

3. Exécuter 20 fois l'algorithme et noter les résultats obtenus. Qui semble gagner le plus souvent la course ?

4. Le nombre caché

Le jeu du nombre caché se joue à deux joueurs. Un des joueurs choisit un nombre entier entre 1 et 100, appelé le nombre caché, et le second joueur doit deviner ce nombre. Après que le second joueur a donné un nombre, le premier l'aide (sans tricher) en lui disant au choix que :

- C'est le bon nombre ; la partie est alors terminée et le second joueur a gagné.
- C'est plus (le nombre caché est plus grand) ;
- C'est moins (le nombre caché est plus petit).

En cas d'échec, le second joueur donne à nouveau un nombre jusqu'à trouver le nombre caché.

On peut limiter (ici à 6) le nombre de tentatives du second joueur pour permettre au premier de gagner.

On donne ci-dessous une version du jeu où c'est l'ordinateur qui choisit un nombre que l'humain doit deviner :

Langage pseudo-algorithmique :	TI :	Casio :
N prend la valeur alea_entier(1,100)	entAleat(1,100)→N	RanInt#(1,100)→N↵
Saisir R	Prompt R	"R=" ?→R↵
T prend la valeur 1	1→T	1→T↵
Tant que N≠R et T < 6 Faire	While N≠R and T < 6	While N≠R And T < 6↵
Si N < R Alors	If N < R	If N < R↵
Afficher "C'est moins"	Then	Then "C'EST MOINS"↵
Sinon	Disp "C'EST MOINS"	Else "C'EST PLUS"↵
Afficher "C'est plus"	Else	IfEnd↵
FinSi	Disp "C'EST PLUS"	"R=" ?→R↵
Saisir R	End	T+1→T↵
T prend la valeur T+1	Prompt R	WhileEnd↵
FinTant	T+1→T	If N=R↵
Si N=R Alors	End	Then "GAGNE!"↵
Afficher "Gagné!"	If N=R	Else "PERDU!"↵
Sinon	Then	"LE NOMBRE ETAIT"↵
Afficher "Perdu ! Le nombre était"	Disp "GAGNE!"	N↵
Afficher N	Else	IfEnd
FinSi	Disp "PERDU!"	
	Disp "LE NOMBRE ETAIT"	
	Disp N	
	End	