Algorithmes

Pour chaque exercices je fais une phase de recherche sur le papier pour préparer le programme, je me demande quelles actions va mener la calculatrice, dans quelle ordre, quelles sont les informations dont elle a besoin, et à quel moment. C'est ainsi que je crée l'algorithme de mon programme, petit texte indiquant en français ce que je vais mettre dans mon programme.

Entrées / Sorties

Exemple 1 : Je veux faire un programme qui va me donner les coordonnées du milieu d'un segment en fonction de celles des extrémités.

Algorithme

Récupérer les informations : (coordonnées des extrémités)	Entrée
Calculer la moyenne des abscisses, ça sera l'abscisse de mon milieu.	Traitement des données
Calculer la moyenne des ordonnées, ça sera l'ordonnée de mon milieu.	Traitement des données
J'affiche les coordonnées de mon milieu.	Sortie

Programme GEOMANA

```
Disp « COORDONNEES DE A »
Prompt X, Y : X→ A : Y→ B
Disp  « COORDONNEES DE B»
Prompt X, Y : X→ C : Y→ D
(A+C)/2→X : (B+D)/2→Y
Disp « LES COORDONNEES DU »,  « MILIEU SONT »,  X►Frac, Y►Frac
```

Remarque : Dans les programmes, vous pouvez écrire l'expression telle qu'elle est proposée sur le papier, quand il est écrit sur la feuille « : » vous pouvez aussi aller à la ligne

A retenir : **Disp** « expression 1», X, « expression 2», … sert à afficher des expressions , et des valeurs
Prompt X, Z, … Sert à demander et stocker les valeurs de différentes variables

Test

Exemple 2 On va créer un programme qui me dira si un nombre est divisible par un autre.

Algorithme

Récupérer le nombre N et le diviseur potentiel D.

On **teste** si la division de N par D tombe juste
Si c'est le cas : on affiche « N est divisible par D »
Sinon : « N n'est pas divisible par D »

Programme DIVISEUR

```
: Disp « N LE NOMBRE »
: Prompt N
: Disp « D LE DIVISEUR POTENTIEL »
: Prompt D
: If R=0
: Then
: Disp « D DIVISE N »
: Else
: partDéc(N/D) →R
: Disp « D NE DIVISE PAS N »
: End
```

A retenir : **If** test : **Then** action 1 : **Else** action 2 : **End** ou **If** test : **Then** action : **End**
Permet de faire une action dans le cas ou le test est concluant, et éventuellement une autre action dans le cas contraire.

partDéc(X) (ou **fPart**(X) en anglais) donne la partie décimale de X

Exemple 3 : On demande à la calculatrice de choisir un nombre au hasard, suivant sa valeur on dira « pile » ou « face ».

commande sur la Ti :

entAléat(A,B) donne un nombre entier entre A et B

Algorithme :

tout est dit dans l'énoncé de l'exercice

Programme PILEFACE

```
: entAléat(0,1)→ X
: If X=0
: Then
: Disp « PILE »
: Else
: Disp « FACE »
: End
```

Boucle

Exemple 4 : on veut simuler N lancers de pièce, et donner les fréquences de piles et de face obtenues.

Algorithme

Initialiser le nombre pile et face
Demander N le nombre de simulations à effectuer (la taille de l'échantillon)
Faire N fois la même séquence :
Simuler un lancer
Augmenter le nombre de face si j'ai obtenu Face — Augmenter le nombre de pile si j'ai obtenu Pile
Une fois la boucle terminée
Diviser le nombre de Face par N, afficher cette fréquence — idem pour les Piles

Programme SIMCOIN

```
0→F
Disp « NOMBRE DE SIMULATIONS »
Prompt N
    For (I,1,N)
    If EntAléat(0,1)=0 : Then : F+1→F : End
    End
Disp « LA FREQUENCE DE », « FACE EST »,F/N
Disp « LA FREQUENCE DE », « PILE  EST»,1-F/N
```

SIMCOIN2

```
{0,0}→L1
Disp « NOMBRE DE SIMULATIONS »
Prompt N
    For (I,1,N)
     EntAléat(1,2)→ X
    L1(X) +1→ L1(X)
    End
Disp « LA FREQUENCE DE »,  « FACE EST », L1(1)/N
Disp « LA FREQUENCE DE », « PILE  EST», L1(2)/N
```

Boucle Conditionnelle

Exemple 5 : on veut faire un jeu de devinette qui ne s'arrête que lorsque l'on donne la bonne réponse. La calculatrice choisi un nombre au hasard entre 1 et 100 et dira « trop fort » ou « trop faible » à chaque proposition du joueur

commande sur la Ti :

While condition
: actions
: actions
: End
crée une boucle qui effectuera ses actions tant que la condition sera vérifiée

Programme DEVINETTE

```
: EntAléat(1,100)→X
: 0→P
: Disp « VOTRE PROPOSITION »
: Prompt P
: While P ≠ X
: If P>X
: Then
: Disp « TROP GRAND »
: Else
: Disp « TROP PETIT »
: End
: Prompt P
: End
```

Il faut prendre le temps de comprendre les programmes, car ils seront votre source d'inspiration pour certains de vos exercices.