

Dans cette troisième leçon de l'Unité 3 vous allez étudier une autre forme de structure **If** en TI-Basic et voir l'utilité de la compréhension des algorithmes numériques.

Objectifs :

- Voir la structure **If...Then...Else...End**.
- Étudier un test permettant de savoir si une valeur est un entier.
- Qu'est-ce qu'un **Algorithme** ?

Indication : Cette leçon introduit le concept d'**algorithme**. Ce principe est essentiel à la réussite des programmes. Bien que nous n'approfondissons pas trop l'approche algorithmique, l'idée d'avoir un plan avant d'écrire un programme est cruciale, puisque l'écriture d'un programme exige d'abord une compréhension de la tâche et la connaissance des commandes de programmation et de leur but. Alors, nous pouvons concevoir un algorithme pour résoudre le problème.

Cette leçon termine l'étude des trois formes de la structure conditionnelle **If** par une discussion sur **If...Then...Else...End**. La syntaxe est importante : chacun des quatre mots-clés apparaît comme des instructions séparées dans le programme et seul **If** peut être suivi de quelque chose sur la même ligne : la *condition*.

À propos du If...Then...Else...End

Dans la précédente activité nous avons étudié la structure **If...Then...End**. Dans certains cas nous aurons à prendre deux différents types d'instructions suivant une condition. La nouvelle structure **If...Then...Else...End** est similaire à :

```
If <condition>  
Then  
  <bloc_vrai>  
Else  
  <bloc_faux>  
End
```

Note :

Then, **Else**, et **End** sont sur des lignes séparées.

Le <bloc_vrai> est l'ensemble des instructions qui seront exécutées quand la <condition> est vraie.

Le <bloc_faux> est l'ensemble des instructions qui seront exécutées quand la <condition> est fausse.

Par conséquent, l'un des deux blocs sera exécuté.



Programmer avec If...Then...Else...End

Nous allons écrire un programme pour voir si le nombre entré est ou n'est pas un carré parfait. Un 'carré parfait' est le carré d'un entier, comme 25 (5^2). La méthode utilisée ici est de prendre la racine carrée du nombre et tester si c'est un entier. Le listing du programme se trouve dans l'écran de droite.

Note :

Utilisez la touche $\sqrt{\quad}$ du clavier pour le symbole racine carré.

If, Then, Else, et End sont tous dans le menu CTL de `[prgm]`.

La fonction **partEnt()** est dans le menu NBRE de `[math]`. La fonction retourne la partie entière, par exemple : `partEnt(6.56) → 6` `partEnt(-2.01) → -3`

Essayez vos propres exemples de **partEnt()** dans l'écran de calcul.

Pensez à fermer les parenthèses pour la fonction **partEnt()** et pour la fonction racine carrée. Vous ne pouvez utiliser que des MAJUSCULES dans vos messages **Disp** sur la calculatrice. Par contre, à l'aide de **TI-Connect CE**, vous pouvez entrer des minuscules.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM: CARRE
:Effécran
:Input "ENTREZ UN NOMBRE :
",N
:If √(N)=partEnt(√(N))
:Then
:Disp "EST UN CARRE PARFAI
T"
:Else
:Disp "N'EST PAS UN CARRE
PROGRAM: CARRE
",N
:If √(N)=partEnt(√(N))
:Then
:Disp "EST UN CARRE PARFAI
T"
:Else
:Disp "N'EST PAS UN CARRE
PARFAIT"
:End
```

Indication : On peut améliorer l'affichage du résultat du programme en affichant dans la phrase le nombre entré. Ceci demandera l'utilisation de **Output()** plutôt que de **Disp**.

Algorithmes

Les techniques comme celles utilisées dans cet exercice sont connues sous le nom d'algorithmes'. Un **algorithme** est une procédure ou formule pour résoudre un problème.

Une recette pour faire cuire un gâteau est un algorithme. Si vous suivez la recette vous aurez un gâteau. Toute formule mathématique, comme la formule donnant l'aire d'un triangle ($A=B*H/2$) est un algorithme : elle vous donne une méthode pour déterminer une nouvelle valeur basée sur des valeurs existantes. Les techniques d'algorithmes tels que le 'carré parfait' au-dessus sont des outils importants pour résoudre des problèmes. L'apprentissage des algorithmes informatiques courants tels que celui-ci enrichira votre expérience de programmation.

Voici la recette d'un gâteau...

1. Préparer la pâte à gâteau selon les instructions sur la boîte.
2. Cuire au four comme indiqué sur l'emballage - en deux couches de gâteau.
3. Laissez refroidir dans les moules pendant 10 minutes.
4. Retirer du four, mettre sur une grille.
5. Laisser refroidir complètement.
6. Battre au fouet pendant 2 minutes le mélange de pouding et le lait.
7. L'étendre immédiatement sur les gâteaux.
8. Empilez les couches de gâteau.
9. Nappez avec la crème fouettée.
10. Dégustez !

Comme dans un programme, le pâtissier suit les étapes du début à la fin. À la fin il y a un gâteau délicieux à déguster. Quand un ordinateur suit les étapes d'un programme (l'algorithme) le résultat désiré est atteint. En fait, il existe une branche de l'informatique qui traite de la preuve qu'un algorithme donnera le résultat désiré. C'est similaire à la démonstration de théorèmes de mathématiques.





Indication : Un exemple d'algorithme informatique :

Pour arrondir un nombre A à son entier le plus proche on peut utiliser :

`partEnt(A+0.5)`

Pourquoi cela marche-t-il ? Quand la partie décimale de A est inférieure à 0.5, ajouter 0.5 au nombre garde le nombre inférieur au premier entier supérieur. Donc la fonction `partEnt()` tronque le nombre de sa partie décimale, laissant juste la partie entière.

Exemples :

A=3.4 A+0.5 = 3.9 `partEnt(3.9) = 3` (arrondi à l'entier inférieur)

A=3.7 A+0.5 = 4.2 `partEnt(4.2) = 4` (arrondi à l'entier supérieur)

Mais bien sûr, la TI-83 Premium CE a aussi une fonction `arrondir()` dans le menu NBRE de `math` !