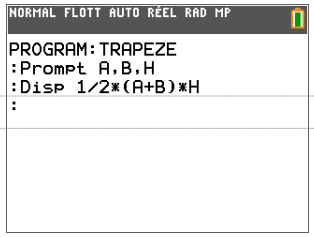
Unité 2 : Utilisation des Variables et des Expressions	Étape 3 : Expressions et stockage
Dans cette troisième leçon de l'Unité 2 vous allez apprendre comment utiliser les expressions et stocker des valeurs dans des variables à l'intérieur de programmes.	**Objectifs:** • Apprendre à programmer des expressions mathématiques. • Comprendre l'ordre des opérations. • Faire la différence entre variables mathématiques et variables de programmes informatiques. • Évaluer des **expressions.** • **Stocker** les résultats d'expressions dans des variables.

Expressions

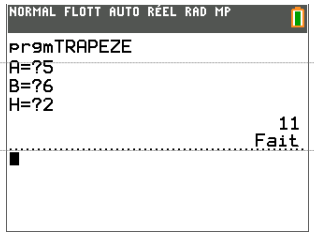
Des éléments tels que **A²** et **A+B** sont appelés *expressions*. Les expressions peuvent être trouvées dans les formules mathématiques. Par exemple, la **formule** donnant l'aire d'un triangle est $A=1/2\times B\times H$. L'**expression** est $1/2\times B\times H$

Un programme évalue une expression en utilisant les valeurs courantes de toutes les variables et donne le résultat sous forme de valeur numérique. Les expressions sont évaluées en utilisant l'*ordre algébrique des opérations*.

Essayez le programme de droite qui calcule l'aire d'un *trapèze* de bases **A** et **B** et de hauteur **H**.



Vous **ne pouvez pas** utiliser des noms de variables tels que **B1** ou **B2**. La calculatrice interprète ceci comme les expressions $B\times 1$ et $B\times 2$ et ceci peut causer une erreur quand c'est utilisé incorrectement. Vous **ne pouvez pas** non plus utiliser des noms de variables comportant plus d'une lettre comme **AB**. Comme ça été dit plus tôt, ceci signifie $A\times B$. Ceci est appelé multiplication *implicite* car le signe de multiplication entre les variables est 'implicite' ou supposé.



Expressions mathématiques et expressions informatiques

Bien qu'il y ait beaucoup de similarités en *apparence* entre les expressions en mathématiques et dans les programmes informatiques, il y a aussi des différences importantes. La plus significative est que dans une expression mathématique les variables ont le statut de nombres 'inconnues' et sont remplacées par des nombres quand c'est nécessaire. Dans une expression informatique les variables sont des *noms* pour des nombres.

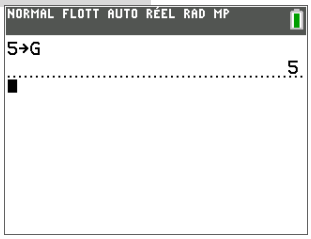
En mathématiques, nous utilisons des formules pour énoncer un rapport entre des choses comme aires et longueurs. Dans des programmes, nous utilisons les expressions pour faire les calculs et les valeurs des variables sont utilisées pour calculer le résultat. Lorsqu'on *écrit* le programme on tape l'expression mais quand on exécute le programme l'expression est calculée et crée un résultat pouvant être utilisé ultérieurement.

Indication : L'une des instructions les plus déroutante dans un programme pour des débutants est x+1→x. C'est à peu près clair dans la syntaxe du TI-Basic que 1 est ajouté à la variable x et ensuite stocké dans la variable x, ainsi le x à gauche et le x à droite représentent des valeurs différentes. Cette instruction est connue comme étant un 'compteur' car à chaque fois qu'elle est effectuée (dans une boucle) la valeur de x augmente de 1. Mais dans d'autres langages comme le B.A.S.I.C et Lua (et bien d'autres) l'instruction apparaît comme x=x+1 ce qui est clairement une assertion fausse au sens mathématique mais parfaitement correcte dans un programme !

Stocker des valeurs dans des variables: L'instruction d'affectation

L'opérateur [sto→] est utilisé pour **stocker** (assigner) le résultat d'une expression dans une variable. Appuyer sur la touche [sto→] affiche le symbole →.

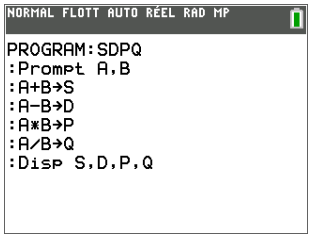
Après avoir appuyé sur [entrer], l'écran de calcul affiche le résultat de l'expression *et* la variable **G** contient maintenant la valeur **5**. La **valeur** ou l'**expression** donnant la **valeur** doit être avant la flèche. Ceci est appelé l'instruction d'*affectation* car elle assigne une valeur à une variable. Le symbole après la flèche doit être une **variable**.



Programmation avec des instructions d'affectation

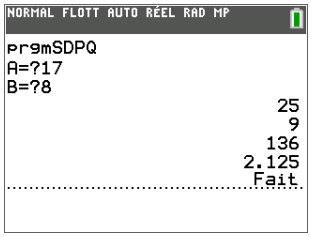
Écrivons un programme qui vous demande d'entrer deux nombres, stocke leur somme, différence, produit et quotient dans quatre variables, puis affiche les résultats.

1. Commencez un nouveau programme. Le nom du programme est **SDPQ**.
2. Prompt pour deux variables **A** et **B**.
3. Stockez les quatre expressions dans quatre *autres* variables **S**, **D**, **P** et **Q**.
4. Affichez **S**, **D**, **P** et **Q**.
5. Lancez le programme.



Entrez une valeur pour **A** et une autre pour **B**.

Assurez-vous d'utiliser des nombres pour lesquels vous pouvez confirmer que les calculs seront effectués correctement ! Ce que vous saurez en 'testant' le programme.



Note : L'instruction d'affectation du TI-Basic ne s'écrit que d'une façon, tout d'abord l'expression, puis l'opérateur 'stocke' et enfin la variable. Ceci rend la lecture facile de gauche à droite. Dans la plupart des autres langages l'ordre est inversé, comme par exemple S=A+B. C'est inversé, car l'ordinateur évalue en premier l'expression à droite, puis affecte la valeur trouvée dans la variable à gauche.

Indication : Il y a de nombreux avantages à stocker des valeurs dans des variables. L'utilisation des variables rend plus aisé l'affichage des résultats. Il permet aussi d'utiliser la variable dans des calculs à venir. Si vous trouvez que vous utilisez le même nombre dans de nombreux endroits dans votre programme, alors vous pouvez le stocker dans une variable et l'utiliser partout où il apparaît dans votre programme.

Apport d'améliorations

Vous pouvez améliorer le programme en utilisant **Output(** à la place de **Disp** pour montrer les valeurs initiales entrées *et* les quatre résultats correctement affichées. Essayez !

À droite vous trouvez un fragment de code et un écran de l'exécution du programme en partie achevé. Il reste encore un peu de travail à faire ici, aussi nous vous laissons terminer le programme.

Pensez à inclure une instruction **Pause** après les instructions **Output** pour empêcher le message 'Fait' qui gâche un peu l'affichage.

Rappelez-vous que les **calculatrices TI-8x** n'ont pas la même taille d'écran de calcul que la **TI-83 Premium CE** (idem pour la taille de l'écran GRAPH) aussi prenez en compte.

L'écran de calcul de la **TI-84 Plus** a 16 caractères par ligne et 8 lignes, alors que celui de la **TI-83 Premium CE** HOME a 26 caractères par ligne et 10 lignes.

```
NORMAL FLOTT AUTO RÉEL RAD MP
PROGRAM:SDPQ
:Prompt A,B
:A+B→S
:A-B→D
:A*B→P
:A/B→Q
:Effécran
:Output(2,2,"A = ")
:Output(2,6,A)
:Output(3,2,"B = ")
```

```
NORMAL FLOTT AUTO RÉEL RAD MP

A = 17
B = 8

   Somme = 25
   Différence = 9
   Produit = 136
   Quotient = 2.125
```

Indication : Expérimentez avec d'autres formules mathématiques et utilisez l'instruction d'affectation pour stocker le résultat pour rendre plus aisé l'affichage des résultats à l'aide de **Disp** ou d'**Output**. Dans l'application de cette Unité trois formules mathématiques sont données à calculer.