

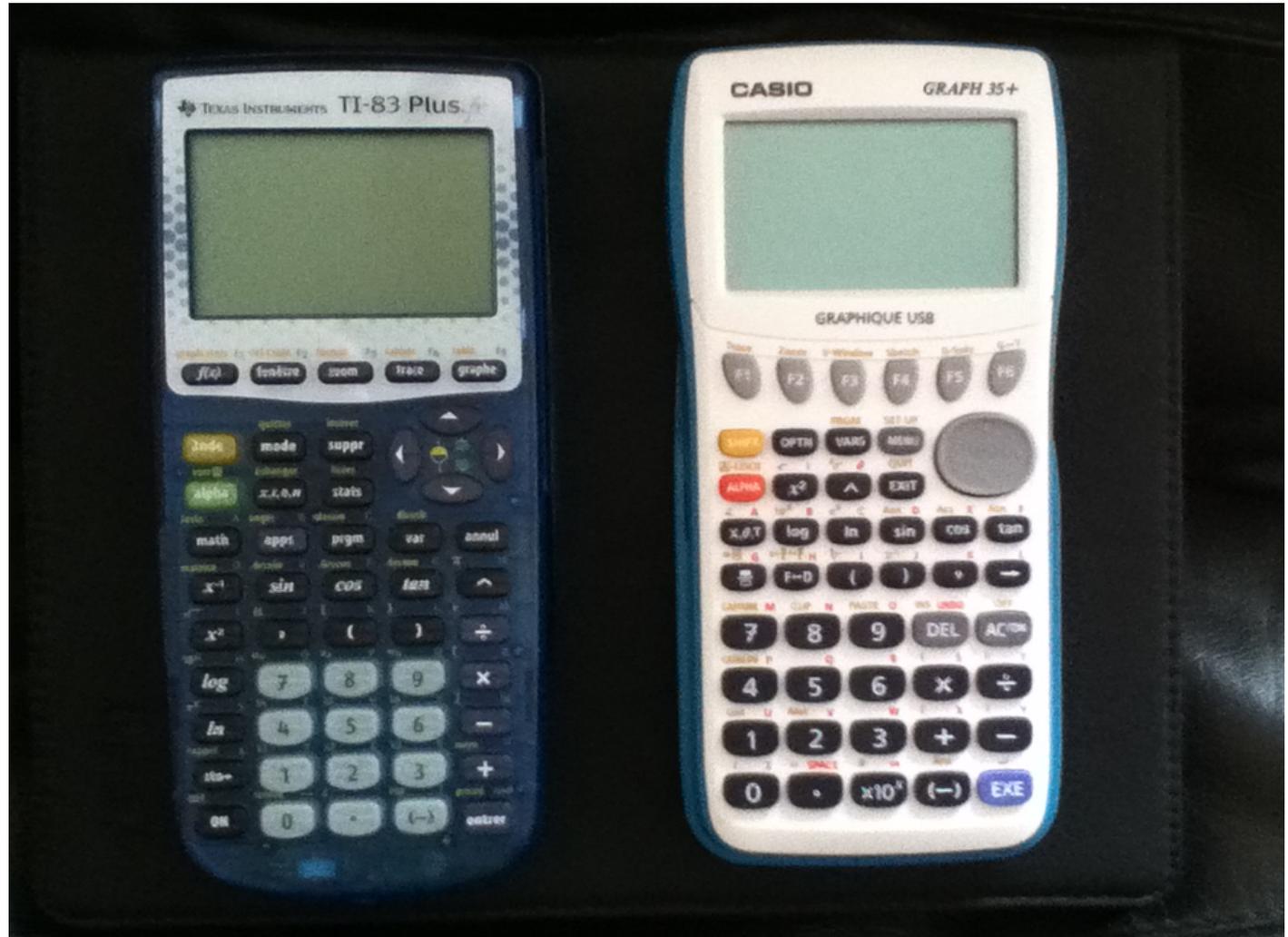


Calculatrices

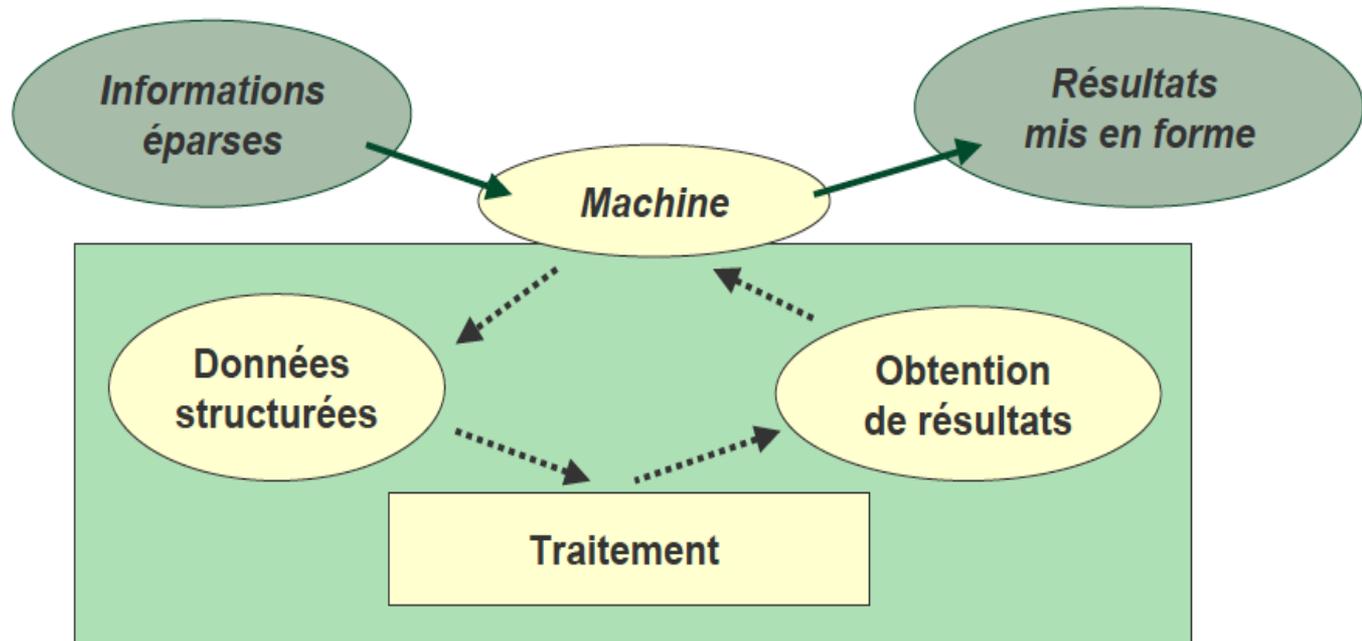
Programmer sa calculatrice

Livret 2

Machines utilisées: TI83+ et GRAPH35+



Algorithme et Programme:



Un **algorithme** = suite d'étapes élémentaires à exécuter.
Un **programme** = algorithme codé dans un langage compréhensible de la calculatrice.

Définitions:

- **Entrée:**

Saisir des valeurs pour qu'elles soient utilisées par le programme.



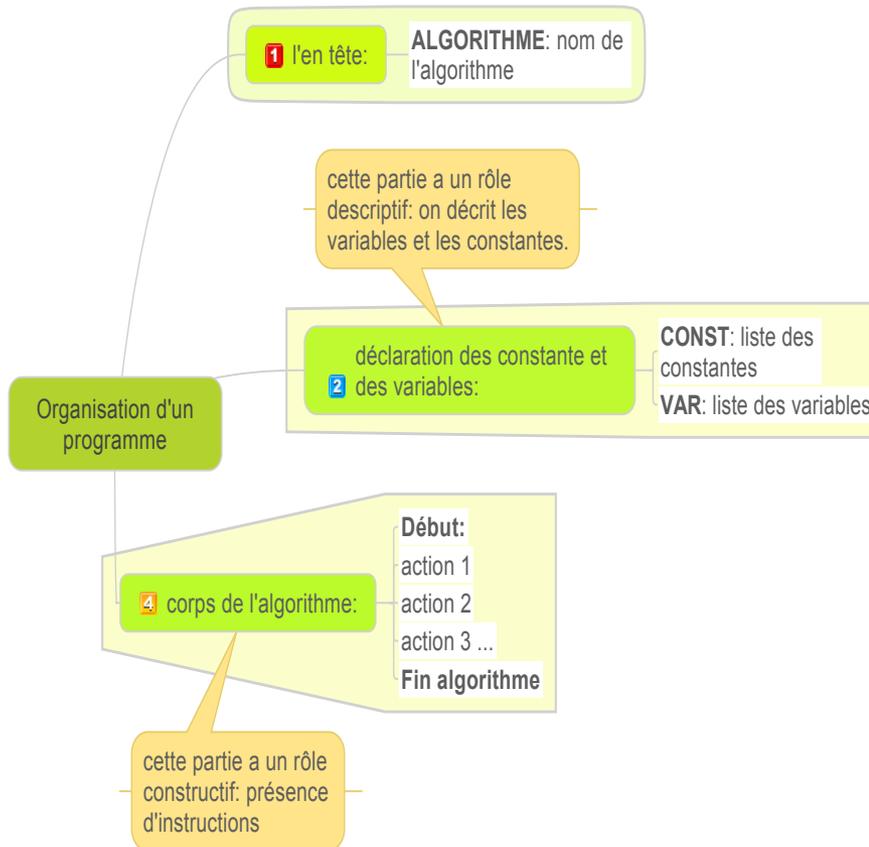
- **Affectation:**

Instruction permettant d'attribuer une valeur à une variable.

- **Sortie:**

Afficher les valeurs des variables après traitement.

Structure d'un programme:



Attention: on écrit les données d'entrée, de sortie ou intermédiaires.
Une donnée de **type simple** ne contient qu'une **seule** information.
Une **constante** est une donnée dont la valeur est précisée au début de l'algorithme, alors qu'une **variable** est une donnée dont la valeur est indéterminée au début de l'algorithme.

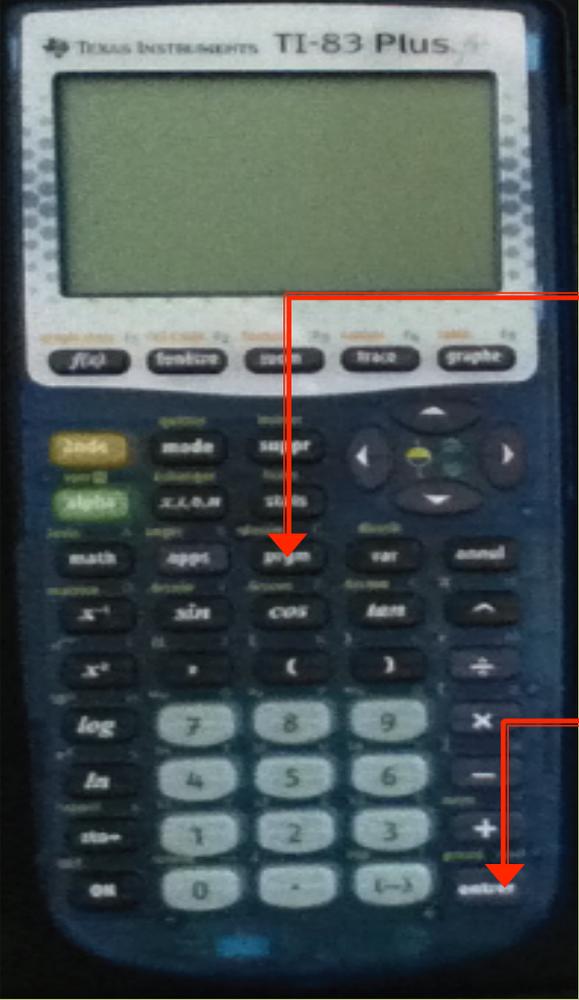
Créer, modifier et exécuter un programme



TI 83+	Graph 35+
<p>On utilise la touche prgm On choisit ensuite:</p> <ol style="list-style-type: none">1. .ENTRER: on exécute le programme2. .EDIT: on peut modifier un programme3. .NOUV: on crée un nouveau programme	<p>On utilise la touche MENU , on sélectionne le menu PRGM et on valide par la touche EXE On utilise les touches F1 à F6 afin d'exécuter, de créer de modifier des programmes</p>

Créer un nouveau programme:

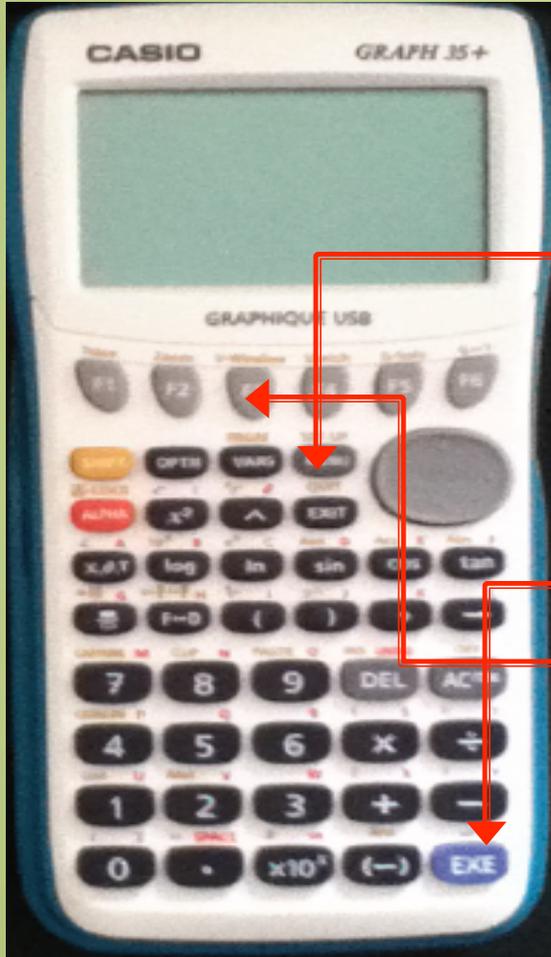


TI 83+	Touches:
 A photograph of a TI-83 Plus calculator. Red arrows point from the text on the right to the 'prgm' key, the right arrow key, and the 'enter' key on the calculator's keypad.	<ol style="list-style-type: none"><li data-bbox="1097 318 1468 368">1. Touche prgm<li data-bbox="1097 539 1680 646">2. À l'aide du curseur, on sélectionne nouv<li data-bbox="1097 825 1719 932">3. On valide avec la touche entrer<li data-bbox="1097 1053 1526 1160">4. Taper le nom du programme

Créer un nouveau programme:

Casio Graph 35+

Touches:



1. Touche **menu**

2. Avec le curseur se placer sur **PRGM**

3. Appuyer sur la touche **EXE**

4. Taper sur la touche **F3**

5. Entrer le nom du programme



Principales instructions: TI 83+



<p><u>Entrée:</u> Saisie du nombre choisi</p>	<p>: INPUT A Ou : PROMPT A Ces deux instructions se trouvent en appuyant sur prgm puis à l'aide des flèches sur E/S .</p>	<p>L'instruction input fera apparaître ? sur l'écran: on tape un nombre qui sera mis dans la case mémoire A. L'instruction prompt est identique sauf qu'elle fait apparaître A=? à l'écran.</p>
<p><u>Traitement:</u> Effectuer le calcul et le mettre dans la case mémoire y</p>	<p>$2(A - 4)^2 \rightarrow y$</p>	<p>On met dans la case mémoire y le résultat . On trouve \rightarrow en utilisant la touche STO.</p>
<p><u>Sortie:</u> Afficher y</p>	<p>:DISP y (prgm puis E/S)</p>	<p>Le résultat va s'afficher sur l'écran.</p>

Exécuter ou modifier un programme:

- *Sortir du mode de programmation:*

Il faut utiliser les touches **2nde mode**

- *Exécuter ce programme:*

Il faut appuyer sur **prgm**, choisir l'instruction **EXEC** et choisir le programme souhaité puis faire **entrer**.

- *Modifier ce programme:*

Il faut utiliser la touche **prgm** et sélectionner grâce au curseur l'instruction **EDIT** et le programme que l'on souhaite modifier. On utilise la touche **entrer** et on modifie la ligne du programme.



Principales instructions: GRAPH 35+



<p><u>Entrée:</u> Saisie du nombre choisi</p>	<p>Taper ? $\rightarrow A$</p> <p>On peut également taper « A= »? $\rightarrow A$ (si plusieurs variable)</p>	<p>Pour obtenir ? On utilise les touches SHIFT VARS F4.</p> <p>Les « » s'obtiennent avec les instructions CHAR puis SYBL.</p> <p>Le signe = est obtenu grâce à SHIFT et à .</p>
<p><u>Traitement:</u> Effectuer le calcul et le mettre dans la case mémoire y</p>	<p>«A=»? $\rightarrow A \downarrow$ $2(A - 4)^2 \rightarrow y \downarrow$</p>	<p>La flèche est obtenue grâce à la touche \rightarrow</p>
<p><u>Sortie:</u> Afficher y</p>	<p>«y=»? : y</p>	<p>Le résultat va s'afficher sur l'écran.</p>

Exécuter ou modifier un programme:

- *Sortir du mode de programmation:*

Il faut utiliser la touche **EXIT** une ou plusieurs fois afin de revenir à la liste des programmes.

- *Exécuter ce programme:*

Il faut sélectionner le programme grâce au curseur et on valide avec l'instruction **EXE**.

On tape sur **F1 (EXE)** pour l'exécuter.

- *Modifier ce programme:*

Il faut utiliser la touche **MENU** puis sélectionner le sous menu **PRGM** et valider grâce à la touche **EXE**.

On choisit le programme à modifier et on tape **F2 (EDIT)** et on modifie la ligne du programme.



Exemples:

■ Exemple 1:

- ✓ Lire un nombre a
- ✓ Lire un nombre b
- ✓ Calculer la moyenne de a et de b
- ✓ Afficher cette moyenne



✓ **Variables:**

A, B des nombres réels

✓ **Initialisation ou entrée des données:**

Saisir A

Saisir B

✓ **Traitement des données:**

Affecter à C la valeur: $(A+B)/2$

✓ **Sortie:**

Afficher «C=», C

TI 83+

```
PROGRAM:MOYENNE
:PROMPT A
:PROMPT B
:(A+B):2→C
:Disp "C=",C
```

GRAPH 35+

```
=====MOYENNE=====
"A=":?→A↓
"B=":?→B↓
(A+B):2→C↓
"C=":C
```

Exemples:

■ Exemple 2:

✓ **Variables:**

A, B, M, P des nombres réels

✓ **Initialisation ou entrée des données:**

Saisir A

Saisir P

✓ **Traitement des données:**

Affecter à M la valeur: $A \times \frac{P}{100}$

$A - M \rightarrow B$

✓ **Sortie:**

Afficher « le montant de la remise », M

Afficher « le prix à payer », B



TI 83+	GRAPH 35+
PROGRAM:SOLDES :PROMPT A,P :A*P/100→M :A-M→B :Disp "montant de la remise",M :Disp "prix à payer",B	=====SOLDES===== ? →A↓ ? →P↓ AxP:100→M↓ A-M→B↓ «montant de la remise=»? : M «prix à payer=»? : B



TI 83+	GRAPH 35+	Commentaire:
:EffEcr (commande obtenue avec les touches: prgm avec le curseur choisir E/S et choisir la ligne 8)	ClrText↓ (commande obtenue avec les touches: 2^{nde} prgm puis F6 et commande CLR (=F1) et choisir Text (=F1)	À noter au début du programme pour effacer l'écran.

Principales instructions:



Algorithme: instructions	TI83+	GRAPH35+
Instruction itérative: Pour ... variant de ... à ...	For(I,1,N) <i>Instruction(s)</i> End	For 1→I To N <i>Instruction(s)</i> Next
Instruction itérative: Tant que ...	While <i>condition</i> <i>Instruction(s)</i> End	While <i>condition</i> <i>Instruction(s)</i> WhileEnd
Instruction conditionnelle: ① Si ... Alors ... ② Si ...Alors ... Sinon...	① If <i>condition</i> Then <i>Instruction(s)</i> End ② If <i>condition</i> Then <i>Instruction(s)</i> Else <i>Instruction(s)</i> End	① If <i>condition</i> Then <i>Instruction(s)</i> IfEnd ② If <i>condition</i> Then <i>Instruction(s)</i> Else <i>Instruction(s)</i> IfEnd

Commandes:



TI 83+

- Instructions if, then, else, for, while, end:
 - ❖ On utilise la touche **prgm** puis dans la rubrique **CTL** on sélectionne l'instruction souhaitée.
- pour trouver FRAC, ABS:
 - ❖ On utilise la touche **math** et **F1** (=MATH) pour **FRAC** et **F2** (=NUM) pour **ABS**.
- Pour trouver >,=,... :
 - ❖ On utilise les touches **2^{nde}** puis **math** ensuite on sélectionne.

GRAPH 35+

- Instructions if, then, else, for, while, end:
 - ❖ On utilise les touches **shift** puis **prgm**; ensuite on utilise la touche **F1** qui correspond à **COM** (c'est à dire commandes), et la touche **F6** permet de choisir la boucle.
- pour trouver ABS:
 - ❖ On utilise les touches **SHIFT catalog** .
- Pour trouver >,=, ... :
 - ❖ On utilise les touches **shift** puis **prgm**; la touche **F6** puis la touche **F3** (= REL) et on choisit.

Structures de contrôle:

① Structure linéaire ou séquentielle:

Suite de traitements à exécuter dans l'ordre énoncé.

② Structure alternative ou conditionnelle:

Deux issues possibles, selon qu'une condition est vraie ou fausse, on effectue un traitement ou un autre.

C'est un test.

③ Structure répétitive ou itérative:

Répétition d'un traitement dans un ordre précis un nombre déterminé ou indéterminé de fois.

C'est une boucle.



Pourquoi?

- Le traitement de données est parfois conditionné et se réalise de manière spécifique.

L'ordre des instructions est primordial.



D'où les deux grands types de structures:

- Les structures conditionnelles: on exécute certaines instructions que sous certaines conditions;
 - Les structures itératives: on répète certaines instructions un certain nombre de fois.
- **A retenir**:
 - **Une boucle permet d'itérer un processus un certain nombre de fois;**
 - **Une condition permet de tester si une expression est vraie ou fausse.**

La structure alternative:

if...then...; if...then...else

- La commande « **if** » demande un test: si la condition imposée par le test est remplie alors on passe à la suite; Sinon (c'est à dire que le test est négatif), on effectue un autre traitement.

1. If: on pose la condition;
2. Then: la condition est remplie, on continue d'exécuter les instructions qui suivent;
3. Else: la condition n'est pas remplie, on va exécuter d'autres instructions;
4. End ou IfEnd: fin de la boucle. Le programme se poursuit.



La structure itérative:

For

- La boucle « **For** »: elle s'emploie quand on connaît le nombre de répétitions à l'avance.

On répète un certain traitement un nombre de fois fixé à l'avance.



- Il faut disposer d'une variable *i* qui va être utilisée par la boucle: cette variable *i* est appelée **compteur**, c'est une variable entière qui est incrémentée de 1 à chaque itération.
 - La boucle n'est plus exécutée dès que le compteur atteint la valeur finale.
- Une fois cette valeur atteinte, l'algorithme se poursuit après l'instruction End.

La structure répétitive:

While

- La boucle « **while** » = « tant que »: tant que la condition est réalisée, la boucle sera exécutée (**ne pas se tromper sur la condition sinon la boucle sera ignorée**).
On ne connaît pas le nombre de répétitions à effectuer (contrairement à la boucle précédente).
- La condition est testée chaque fois en début de boucle: dès que la condition de début est mise en défaut, on sort de la boucle et l'algorithme se poursuit après l'instruction End ou WhileEnd.



Exemples:

■ Exemple 1:

Un directeur de salle de cinéma facture chaque séance 9,50 euros. Il a mis en place un système de fidélisation en mettant les réductions suivantes:

1. Les spectateurs qui vont au moins 3 fois par mois au cinéma dispose d'une réduction de 2 euros par séance.
2. Ceux qui vont au cinéma 6 fois ou plus par mois, auront un tarif préférentiel de 5 euros.

Construire un algorithme permettant de calculer le prix total à payer par mois en fonctions du nombres de séances mensuelles, puis le programmer.



- Soit N le nombre de séances mensuelles

protocole

Variables

Les variables utilisées sont des nombres

Entrée

Saisie des variables

Traitement

Opérations sur les variables

Sortie

Affichage des résultats

algorithme

Variables:

N: entier;

Montant: réel;

Entrée:

Saisir N

Traitement:

Si $N < 3$ alors

$9,5 \times N \rightarrow$ Montant;

sinon

Si $N < 6$ alors

$7,5 \times N \rightarrow$ Montant;

sinon

$5 \times N \rightarrow$ Montant;

FinSi

Sortie:

Afficher Montant



TI 83+

GRAPH 35+

```

PROGRAM:CINEMA
:EffEcr
:PROMPT N
:If N<3
:Then
:9,5xN→M
:Else
:If N<6
:Then
:7,5xN→M
:Else
:5xN→M
:End
:End
:Disp "Montant mensuel à
régler", M

```

```

=====CINEMA=====
ClrText
?→N↓
If N<3↓
Then 9.5xN→M↓
Else If N<6↓
Then 7.5xN→M↓
Else 5xN→M↓
IfEnd↓
IfEnd↓
«Montant mensuel à
régler=» : M

```



Remarque importante:

Sur la **TI 83+** :

- On ne doit pas rajouter d'espace après une instruction.

Après une instruction **THEN** ou **ELSE** par défaut pas d'espace: il faut *impérativement* aller à la ligne.

(Ce n'est pas le cas sur la Graph 35+.)



Exemples:

■ Exemple 2:

Pour un polynôme donné, le programme doit afficher le discriminant D et les solutions éventuelles.

Le programme doit demander:

- A
- B
- C

Puis affecter la valeur $B^2 - 4 \cdot A \cdot C$ à D

Afficher D

Si $D < 0$ alors afficher « pas de solution »

Si $D = 0$ alors afficher « une solution: » $X_0 = -B/2A$

Si $D > 0$ alors afficher « 2 solutions réelles distinctes: »

$X_1 = (-B - \text{racine}(D))/(2A)$ et $X_2 = (-B + \text{racine}(D))/(2A)$



TI 83+

```

PROGRAM:DELTA
:EffEcr
:PROMPT A
:PROMPT B
:PROMPT C
:B^2-4xAxC→D
:Disp"D = "
:Disp D
:If D=0
:Then
:Disp"une racine double"
:Disp (-B)/(2A)►Frac
:Else
:If D<0
:Disp "pas de racine"
:Else
:Disp "deux racines réelles
distinctes"
:Disp (-B-√D)/(2A)►Frac
:Disp (-B+√D)/(2A)►Frac
:End

```

GRAPH 35+

```

=====DELTA=====
ClrText↓
"A?":?→A↓
"B?":?→B↓
"C?":?→C↓
B^2-4xAxC→D↓
"D=":D ▲
If D=0↓
Then "1 solution" ↓
"X=":(-b)/(2xA) ↓
Else If D<0↓
Then "pas de solution" ↓
Else "2 solutions" ↓
"X1=":(-B-√D):(2xA)▲
"X2=":(-B+√D):(2xA) ↓
IfEnd↓
IfEnd

```



Exemples:

- Exemple 3:

Clara verse sur un compte 1000 euros.

Ce compte rapporte 2% d'intérêts par an et ces intérêts sont réinvestis chaque année.

On cherche à connaître le nombre d'années au bout duquel le capital est doublé.



Soit S la somme de départ: on l'initialise à 1000 euros.

Soit n le nombre d'années: on l'initialise à 0.

D'une année sur l'autre, la somme est multipliée par 1,02; on le fait tant que la somme n'a pas atteint 2000 euros.



TI 83+

```
PROGRAM:DOUBLE
:EffEcr
:PROMPT S
:PROMPT N
:While S<2000
:1,02*S→S
:N+1→N
:End
:Disp "NBRE ANNEES"
:Disp N
```

GRAPH 35+

```
=====DOUBLE=====
ClrText↓
0→N↓
1000→S↓
While S<2000↓
1,02*S→S↓
N+1→N↓
WhileEnd↓
«NBRE ANNEES»? : N
```

Exemples:

- Exemple 3: la dichotomie

On considère la fonction :

$$f(x) = x^3 + 2x - 2 \text{ sur l'intervalle } I = [-2; 3]$$

On se propose de déterminer une valeur approchée aussi précise que possible de la solution de l'équation $f(x) = 0$.

on va utilisée une méthode qui est basée sur l'utilisation des tableaux de valeurs de la calculatrice pour donner une valeur approchée de la solution de $f(x) = 0$.



■ Explications:



Algorithmme:	Commentaires:
<p><u>Variables:</u> a, b: nombres réels → f: fonction → e: nombre réel → m: nombre réel →</p> <p><u>Entrées:</u> Saisir a, b, f, e</p> <p><u>Traitement:</u> Tant que $b - a > e$ m prend la valeur $\frac{a+b}{2}$ → Si $f(m)$ et $f(a)$ sont de mêmes signes →</p> <p>a prend la valeur m → sinon b prend la valeur m → Fin tant que</p> <p><u>Sortie:</u> Afficher a, b →</p>	<p>Bornes de l'intervalle Fonction étudiée Précision Valeur approchée de la solution</p> <p>On se place au centre de l Si $f(m)$ et $f(a)$ sont de mêmes signes, alors la solution de l'équation $f(x)=0$ est située dans l'intervalle $[m; b]$ On se place sur $[m; b]$</p> <p>On se place sur $[a; m]$</p> <p>On affiche un encadrement de la solution de $f(x)=0$</p>

TI 83+

```

PROGRAM:DICHOT
:EffEcr
:PROMPT A
:PROMPT B
:PROMPT E
:While abs(A-B)>E
:(A+B)/2→C
:If Y1(A)*Y1(C) ≤ 0
:Then
:C→B
:Else
:C→A
:End
:End
:Disp "A"
:Disp A
:Disp "B"
:Disp B
    
```

GRAPH 35+

=====DICHOT=====

ClrText↓

?→A↓

?→B↓

?→E↓

Y=Type↓

"FONCTION" ↓

?→Y1↓

While Abs(B-A)>E↓

(A+B)/2→C↓

If Y1(A)*Y1(C) ≤ 0↓

Then C→B↓

Else C→A↓

IfEnd↓

WhileEnd↓

A▲

B

Pour noter **Y=Type**:
Touche **F4** puis **F4** (=GRPH)
et touche **F3** (=TYPE) et
ensuite **F1** (=Y=)

Pour noter **Y1**:
Touche **VAR**
puis **F4** (=GRPH)
et **F1** (=Y)



Exemples:

- Exemple 4:

Le programme suivant calcule les termes successifs d'une suite arithmétique, après avoir donné le premier terme, la raison ainsi que le nombre de termes souhaités.



Algorithme:

```
Demander: U0
           la raison r
           le rang n du dernier terme souhaité
Pour i allant de 1 à n
Affecter la valeur U+r à U
Afficher U
```

TI 83+

```
PROGRAM:SUITE
:EffEcr
:Disp U0
:INPUT U0
:PROMPT R
:PROMPT N
:For(I,1,N)
:U+R→U
:Disp U
:Pause
:End
```

GRAPH 35+

```
=====SUITE=====
ClrText
"U0?":?→U↓
"R?":?→R↓
"N?":?→N↓
For 1→I to N↓
U+R→U↓
U ▲
Next
```

Commentaire:

Pause: permet d'arrêter le défilement des valeurs, on appuie sur la touche **entrer** pour obtenir la suivante

Commentaire:

Next: fin de l'instruction « for »: passage à la valeur suivante tant que I n'aura pas atteint N. On l'obtient avec les touches **shift prgm** puis **F1**, avec le curseur choisir la boucle **FOR** et la touche **F4**



Exemples:

- Exemple 5:

Le programme suivant calcule l'image d'un nombre par une fonction f.



Algorithme:

Variables:

X: entier

Y: réel

A: entier

B: entier

Début:

pour X allant de A à B faire

$Y \leftarrow 2 * X^2 + X - 3$

Afficher ("X";"Y")

finpour

Fin



TI 83+

```
PROGRAM:IMAGE
:EffEcr
:Prompt A
:Prompt B
:For (X,A,B)
:2*X^2+X+3→Y
:Disp "(,X,,",Y,)"
:Pause
:End
```

GRAPH 35+

```
=====IMAGE=====
ClrText
"A?":?→A↓
"B?":?→B↓
For A→X to B↓
2*X^2+X+3→Y↓
"X=":X▲
"Y=":Y▲
Next
```