

Énoncé

Une entreprise fabrique des microscopes électroniques. Chaque mois, elle produit un nombre x de microscopes compris entre 1 000 et 3 000. Le coût de production, exprimé en euros, de x microscopes est donné par la fonction du 2nd degré :

$$f(x) = 0,448x^2 - 1\,452,7x + 1\,455\,391$$

Chaque microscope est vendu 249 € par l'entreprise. On suppose que l'entreprise parvient à vendre toute sa production.


1. L'entreprise réalise-t-elle un bénéfice lorsqu'elle fabrique et vend 1 200 microscopes ? 2 300 microscopes ?
2. On considère la fonction **Python** suivante :


```
def resultat(x):
    c=0.448*x**2-1452.7*x+1455391
    r=249*x
    return(r-c)
```


Quel est le rôle de cette fonction ?

3. Tester cette fonction avec les valeurs de la première question.

1. Calcul d'image

On commence par saisir l'expression $f(x)$ dans Y_1 du menu , mais également l'expression $249x$ dans Y_2 .

Pour calculer les images de 1 200 et de 2 300 par f , nous allons utiliser le menu  de la calculatrice.

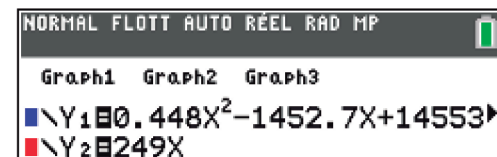
Dans ce menu, dans l'onglet **VAR Y**, nous sélectionnons la commande **1:Fonction**. Nous accédons alors aux noms des différentes fonctions saisies dans la calculatrice (les mêmes que dans le menu ).

Nous saisissons alors le calcul $Y_2(1200) - Y_1(1200)$.

On obtient $-58\,471$ €. L'entreprise ne réalise donc pas de bénéfice.

On procède de même pour **2 300** et on obtient un bénéfice de $88\,599$ €.

Remarque : pour le 2nd calcul, on peut utiliser les touches   pour récupérer le calcul précédent et modifier uniquement la valeur **1 200**.



2. Fonction Python

Le rôle de cette fonction est de calculer, en fonction de la valeur x entrée par l'utilisateur :

- Le coût de production c des x microscopes ; on reconnaît ici la fonction trinôme du 2nd degré f .
- La recette r des x microscopes, qui est une fonction linéaire.
- Le bénéfice réalisé en réalisant la différence des 2 fonctions précédentes.

Dans cette partie, nous allons utiliser le module **Python** disponible dans la calculatrice pour coder la fonction **resultat** dans un script, que nous exécuterons dans la console (**Shell**) afin de réaliser les tests de la 3^{ème} question.


Le module **Python** est accessible via la touche , en sélectionnant la commande **2:Python App**.


On parvient alors dans le « Gestionnaire de scripts », comme indiqué dans le bandeau supérieur. Nous allons donc créer un nouveau script à l'aide de l'onglet **Nouv**, accessible avec la touche **f3**.



A présent, on nomme notre script **MICRO** à l'aide de l'alphabet disponible sur les touches de la calculatrice. Notez que celle-ci est déjà en « mode alphabet », indiqué par le **A** en haut à droite de l'écran. On peut définir le style de notre script avec la touche **f3** ; le choix **2:Calculs Mathématiques** est une option qui permet de charger automatiquement la librairie mathématique de **Python**, qui donne accès aux fonctions usuelles (racine carrée, fonctions trigonométriques, etc.). On valide alors le nom du script par **Ok**.

Nous voici maintenant dans l'éditeur du script **MICRO**, comme indiqué dans le bandeau supérieur.

Dans l'onglet **Fns...** (touche **f1**), on accède immédiatement au menu **Fonc**, où se trouve les commandes nécessaires pour toute fonction.

On sélectionne ainsi **1:def fonction()**: et, dans l'éditeur de script, on saisit immédiatement le nom de notre fonction, le curseur étant déjà correctement placé en mode insertion, à l'aide de la touche .

En appuyant sur la touche , on se retrouve sur la ligne inférieure. Notez que l'indentation, si importante dans le langage **Python**, est déjà implémentée.

Des raccourcis ont été définis sur la calculatrice, pour faciliter le codage. Ainsi, le symbole **=** est accessible directement, via la touche . La touche  quant à elle, permet de coder le carré (****2** en **Python**).

Pour la dernière ligne, rappelons que **return** est accessible via **Fns...**

Une fois notre script terminé, on l'exécute (**Exéc**, via la touche **f4**).

Remarques :

- Le mode « insertion » peut conduire à des erreurs de saisie par les débutants. Toutefois, la prise en main est rapide. Prenez votre temps lors de ce premier script !
- Attention à ne pas oublier les symboles de multiplication entre les nombres et la variable x . C'est une erreur très courante...

3. Tests

A présent, nous nous retrouvons dans la console Python : le Shell.

Notre fonction **resultat** est disponible à l'aide de la touche .

Une fois sélectionnée, nous pouvons la tester avec les valeurs 1 200 et 2 300 de l'énoncé.

Nous retrouvons bien les valeurs calculées précédemment.

```
NORMAL FLOTT AUTO RÉEL RAD MP
Programming
1:TI-Basic
2:Python App
```

```
GESTIONNAIRE DE SCRIPTS
NOUVEAU SCRIPT
Nom=MICRO

Autorisé
- Jusqu'à 8 caractères
- Premier caractère:AàZ
- Caractères restants:AàZ 0à9 _

Calculs Mathématiques
Échap Types Ok
```

```
ÉDITEUR : MICRO
LIGNE DU SCRIPT 0003
# Calculs Mathématiques
from math import *
```

```
ÉDITEUR : MICRO
Fonc Ctl Ops List Type E/S Modul
1:def fonction():
2:return
```

```
ÉDITEUR : MICRO
LIGNE DU SCRIPT 0006
# Calculs Mathématiques
from math import *
def resultat(x):
**c=0.448*x**2-1452.7*x+1455391
**r=249*x
**return (r-c)_

Fns... | a A # | Outils | Exéc | Script
```

```
PYTHON SHELL
>>> # Shell Reinitialized
>>> # L'exécution de MICRO
>>> from MICRO import *
>>> resultat(1200)
-58471.0
>>> resultat(2300)
88599.0
>>> |

Fns... | a A # | Outils | Éditer | Script
```