

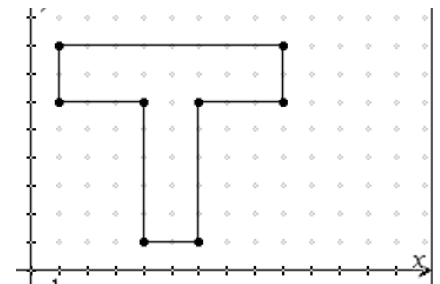


Dans cette leçon, vous allez écrire un programme pour créer une forme bidimensionnelle prédéfinie. Vous allez utiliser les listes dans le programme, ainsi qu'une boucle afin de dessiner les points sur le papier.

Objectifs :

- Définir des listes et utiliser des couples de coordonnées stockées dans celles-ci.
- Dessiner sur le papier une forme à deux dimensions et la reproduire sur la calculatrice.
- Laisser le Rover dessiner la forme sur le papier ou simplement suivre le modèle.

Ce projet nécessite l'utilisation de deux listes qui représentent les coordonnées x et y d'une forme de votre conception. Dans cette leçon, nous allons dessiner un bloc représentant la lettre T comme illustré ci-contre. Notre objectif est de faire en sorte que le Rover dessine cette lettre en utilisant un stylo feutre ou suive le chemin si aucun feutre n'est disponible.



1. Tout d'abord, créer deux listes contenant les paires de coordonnées. Ces listes représentent les coordonnées du bloc de la lettre T. Les abscisses sont dans la liste **xs** et les ordonnées dans la liste **ys**. Si vous le souhaitez, vous pouvez créer un bloc lettre différent.

Optionnel : Paramétrer une représentation graphique des données afin de vous assurer que celle-ci correspond bien à ce que vous souhaitez.

	A xs	B ys	C
=			
1	4	1	
2	6	1	
3	6	9	
4	9	9	
5	9	11	
6	1	11	
7	1	9	
8	4	9	
9	4	1	
10			

Conseil à l'enseignant : Le programme peut aussi entre les coordonnées dans les listes. Utiliser les instructions montrées ci-dessous à inclure dans le programme.

`{1,2,3,4,5}→xs`

`{6,7,8,9,10}→ys`

2. Maintenant nous pouvons écrire un programme qui commandera au Rover le dessin de la forme. Commencer le programme de manière habituelle, éventuellement, fixer l'unité M/UNIT à une valeur plus petite afin que le graphique ne soit pas trop grand. Sélectionner **prgm > Hub > Rover (RV)... > RV Setup... > Send("SET RV.GRID.M/UNIT.**

Souvenez-vous que 0.01 fixe l'unité à 1 cm. Si vous voulez que l'unité soit de 1 pouce, utiliser 0.0254. Ajouter les guillemets à la fin de l'instruction ainsi que la parenthèse fermante.

Send "SET RV.GRID.M/UNIT .0254"





10 Minutes de Code

TI-NSPIRE™ CX AVEC LE TI-INNOVATOR™ ROVER

3. Nous utilisons une boucle **For** pour traiter chaque élément des deux listes. Souvenez-vous que la boucle **For** exige trois arguments, la variable **i** qui contrôle la boucle, commence à **1**, et termine à, **dim(xs)**. **EndFor** est nécessaire à la fin du corps de la boucle.

For i, 1, dim(xs)

EndFor

Note : **dim(xs)** est le nombre d'éléments de la liste **xs**.

4. Le corps de la boucle (le code à l'intérieur de la boucle **For** est constituée d'une seule instruction indiquant au Rover d'atteindre chaque point dans l'ordre.

Send "RV TO XY eval(xs[i]) eval(ys[i])"

5. Après avoir entré le code, faire fonctionner le programme et observer si la forme est correctement dessinée.

UNITE 6 : COMPETENCE 3

NOTES DU PROFESSEUR

```
1.1 *Classeur RAD 7/7
* rover63
Dessine une forme à partir de points dont
les coordonnées sont sauvegardées dans les
listes xs et ys
Send "CONNECT RV"
Send "SET RV.GRID.M/UNIT 0.01"
Text "Appuyer sur Enter pour commencer"
For i,1,dim(xs)
EndFor
EndPrgm
```

```
1.1 *Classeur RAD 6/7
* rover63
Dessine une forme à partir de points dont
les coordonnées sont sauvegardées dans les
listes xs et ys
Send "CONNECT RV"
Send "SET RV.GRID.M/UNIT 0.01"
Text "Appuyer sur Enter pour commencer"
For i,1,dim(xs)
Send "RV TO XY eval(xs[i]) eval(ys[i])"
EndFor
EndPrgm
```