Unité 5 : Utiliser la bibliothèque TI System | **Compétence 1 : Travailler sur des données**

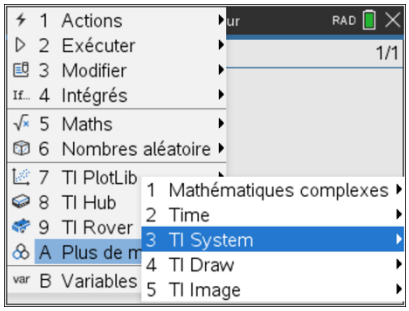
Dans cette première leçon de l'unité 2, vous allez découvrir comment utiliser la bibliothèque **Ti System** pour importer ou exporter des listes dans un script Python.

Objectifs :

- Importer-exporter des listes.
- Réinvestir les notions de l'unité 4 sur les représentations graphiques.

La bibliothèque ou module T**i System** utilisée seule ou en complément des autres, permet de communiquer (dans les deux sens) avec la calculatrice graphique.

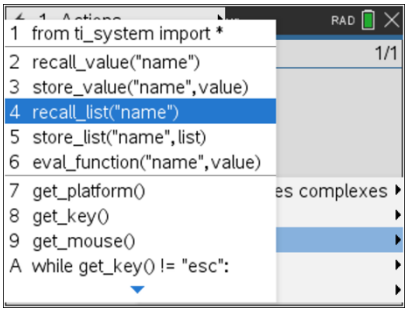
Pour charger cette bibliothèque, appuyer sur la touche [menu] puis **A Plus de modules** et enfin **3 TI System**.



Dans cette leçon, nous allons concentrer notre attention sur l'utilisation des instructions :

4 : var=recall_list(« nom ») et **5 : store_list(« nom »,var)**

Les autres options de cette bibliothèque seront abordées dans les autres leçons de cette unité.

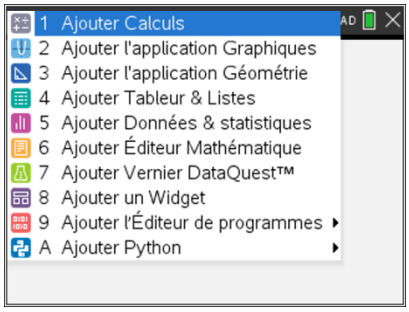


1 : Importer des données depuis la calculatrice.

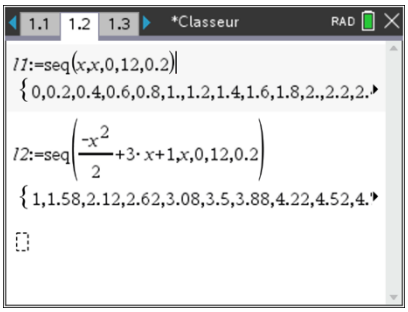
a) Création de deux listes.

Dans un premier temps nous allons simplement créer deux listes contenant les données.

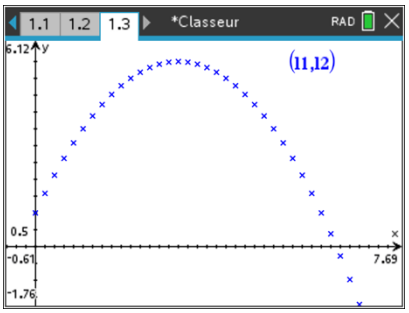
- Créer une nouvelle page de calculs [ctrl] [I] puis choisir **1 Ajouter Calculs.**
- Le même travail est réalisable à partir de l'application **Tableur&Listes**.



Vous allez créer dans la liste $\mathbf{l_1}$ une suite de nombre entre 0 et 12 avec un pas de 0.2 .
Puis une liste $\mathbf{l_2}$ correspondant à la liste des images de x (parcourant la liste $\mathbf{l_1}$ par la fonction f définie par : $x \longmapsto -x^2/2 + 3x + 1$



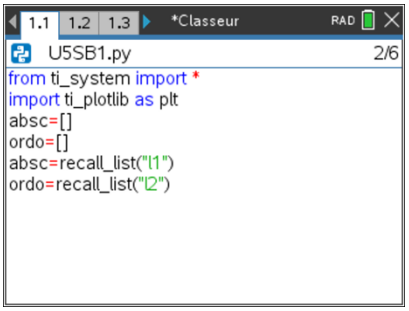
- Insérer une nouvelle page « **Application graphique** ».
- Représenter graphiquement le nuage de points ($\mathbf{l_1}$,$\mathbf{l_2}$).



- Régler les paramètres de la fenêtre graphique tels que : X_{min} = -1.2 ; X_{max} = 8 ; Y_{min}= -0.5 et enfin Y_{max}= 6.5.

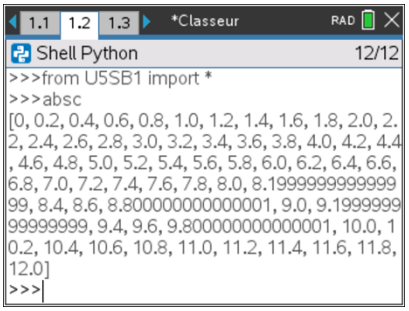
b) Import des données dans un script Python.

- Commencer un nouveau script et le nommer U5SB1.
- A partir de menu importer la bibliothèque **TI System**.
- Créer deux variables listes (vides) **absc** et **ordo**.
- Importer les bibliothèques **TI System** et **TI PlotLib** (il n'y a pas d'ordre particulier à respecter).
- Créer une variable **absc** puis, à partir des options de la bibliothèque **TI System**, choisir l'option **4 : var=recall_list(« nom »)**. Comme les abscisses sont dans la liste $\mathbf{l_1}$, le champ « nom » est complété par le nom complet de la liste.
- Créer une autre variable **ordo** et procéder de la même manière avec la liste $\mathbf{l_2}$.

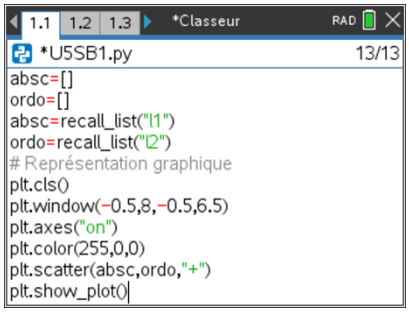


Conseil à l'enseignant : La création de listes vides absc=[ ] et ordo=[ ] n'est pas indispensable, car elles seront créés lors du rappel des listes $\mathbf{l_1}$ et $\mathbf{l_2}$. Cependant il est préférable de conserver les bonnes habitudes apprises lorsque nous ne faisons pas appel au module **TI System** qui nécessitent cette création préalable.

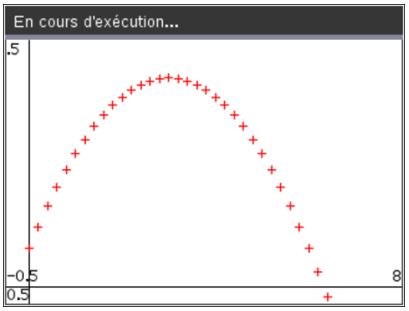
- Exécuter le script, puis vérifier le contenu de vos variables **absc** et **ordo** en appuyant sur la touche [var].
- Rappeler ensuite le nom de la « liste » **absc**, puis valider [enter].
- Procéder de la même façon avec la liste des ordonnées (**ordo**).



c) Représentation graphique.
Paramétrer votre représentation graphique comme proposé sur l'écran ci-contre.



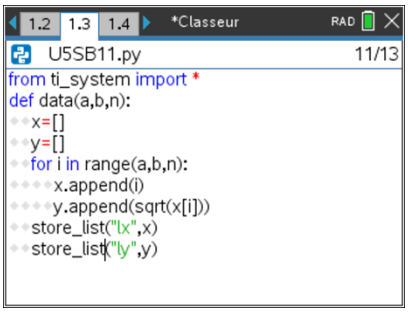
Exécuter votre script [ctrl] [R].



2 : Exporter des données

Créer un nouveau script et le nommer U5SB11.

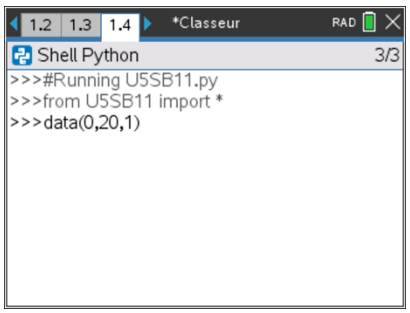
Conseil à l'enseignant : Placer le curseur à la fin d'une ligne et valider. L'ordre d'écriture de l'importation des modules est sans importance.



- Créer une fonction nommée **data(a,b,n)**.
- Vous allez créer deux listes de données, à représenter sous forme d'un nuage de points, comportant des valeurs dans un intervalle **[a ; b],** calculées avec un pas **n**.
- Dans la liste **y**, on calcule la racine carrée des valeurs de la liste **x**.
- Pour créer ces listes de données, nous allons construire une boucle fermée après avoir bien entendu créé deux listes vides.

Conseil à l'enseignant : La création de deux listes vides évite le renvoi d'un message d'erreur lors de l'exécution du script.

Remarque : Attention à l'indentation, les instructions **store_list** n'ont pas à être dans la boucle. Utiliser [del] afin de supprimer un niveau de décalage.



- Exécuter votre script. On prend ici 21 valeurs de 0 à 20 par pas de 1
- Sortir de l'environnement Python et afficher la représentation graphique de vos listes **lx** et **ly** (nuage de points dans l'application graphique par exemple).

Conseil à l'enseignant : l'export vers les listes de la calculatrice sera particulièrement intéressant pour représenter des données acquises par l'intermédiaire de capteurs avec le Microcontrôleur **TI-Innovator™** & **Robot TI-Innovator™ Hub**.