Unité 5 : Utiliser la librairie ti_system | **Compétence 1 : Travailler sur des données**

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

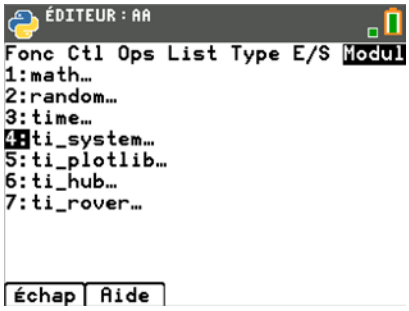
Objectifs :

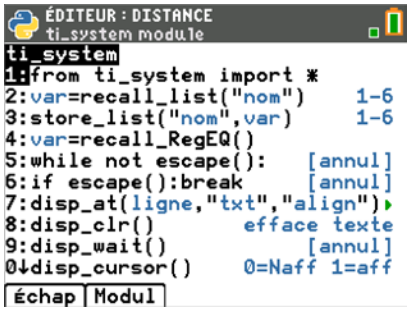
- Importer-exporter des listes.
- Réinvestir les notions de l'unité 4 sur les représentations graphiques.

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

2 : var=recall_list(« nom ») et **3 : store_list(« nom »,var)**

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



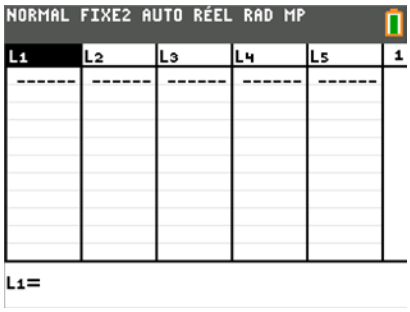


1 : Importer des données depuis la calculatrice.

a) Création de deux listes

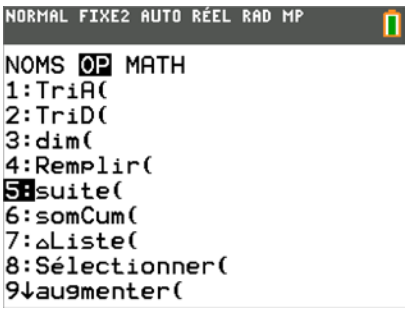
Dans un premier temps nous allons simplement entrer des données dans des listes (L_1 et L_2).

- La calculatrice étant en fonctionnement habituel (hors programmation), effacer le contenu des listes en appuyant sur 2nde + puis 4 : EffTtesListes.
- Entrer dans l'éditeur de liste et mettre le curseur sur L_1 comme sur l'écran ci-contre.

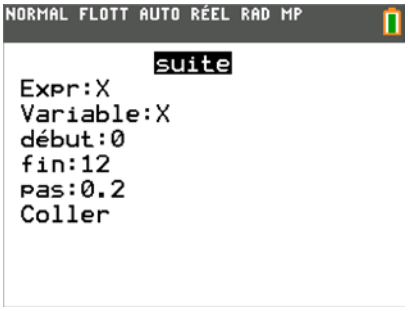


Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

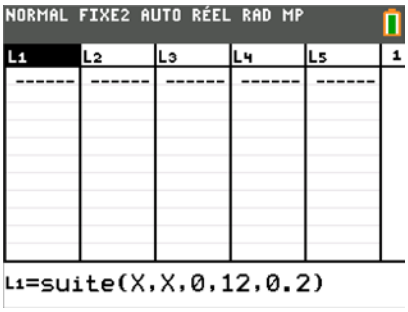
Vous allez créer dans la liste L_1 une suite de nombre entre 0 et 12 afin de pouvoir représenter graphiquement les points calculés, dont on a calculé les coordonnées à l'aide d'une fonction définie par : $x \longmapsto -x^2/2 + 3x + 1$



Appuyer sur [2nde] [stats] [▸] [▸] puis choisir l'instruction **5 : suite** et compléter les paramètres afin de construire une suite de nombres dans l'intervalle $[0\,;\,12]$ par pas de 0.2.

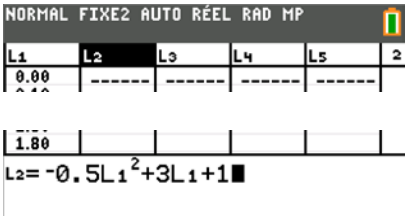


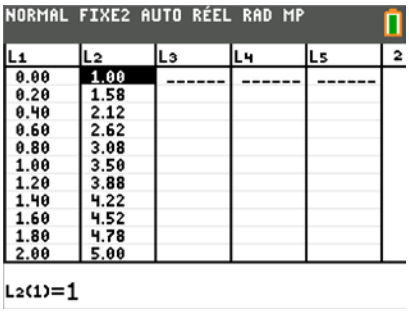
Une fois le curseur sur Coller, appuyer sur la touche [entrer] afin de valider la création de la suite.



Les valeurs calculées sont recopiées dans la liste L_1.

Placer le curseur sur L_2 et demander le calcul des images par la fonction : $x \longmapsto -x^2/2 + 3x + 1$ des valeurs de la liste L_1 à l'aide de l'instruction L_2=-0.5L_1^2+3L_1+1, puis valider le calcul [entrer].

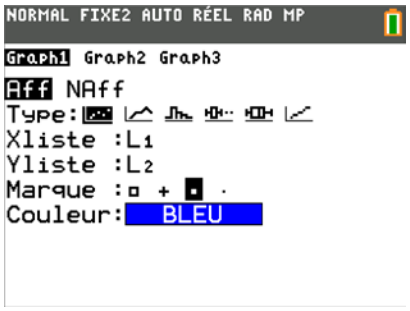


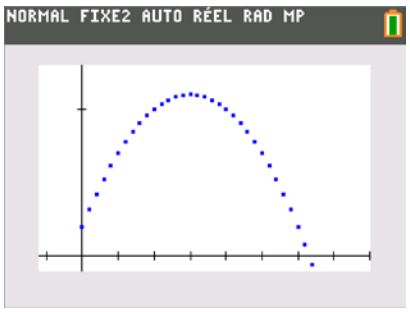


Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

Paramétrer votre représentation graphique sous forme d'un nuage de points. Afficher la représentation graphique zoom 9 en vérifiant par ailleurs qu'aucune fonction n'est activée (touche f(x)).

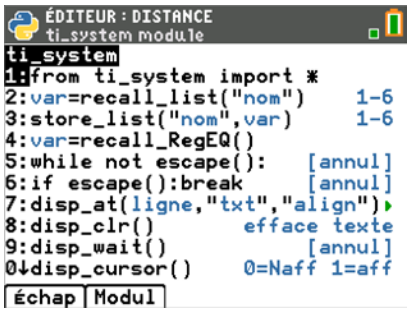
Régler les paramètres de la fenêtre graphique tels que : X_{min} = -1.2 ; X_{max} = 8 ; Y_{min}= -0.5 et enfin Y_{max}= 6.5.





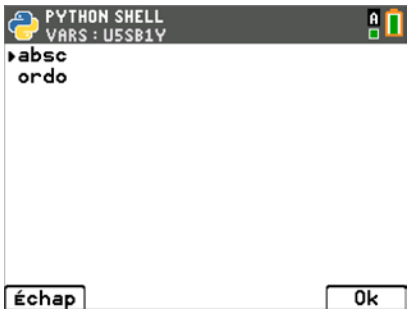
b) Import des données dans un script Python

- Commencer un nouveau script et le nommer U5SB1.
- Créer deux variables listes (vides) **absc** et **ordo**.
- Importer les librairies **ti_system** et **ti_plotlib** (il n'y a pas d'ordre)
- Créer une variable **absc** puis, à partir des options de la librairie **ti_system**, choisir l'option **2 : var=recall_list(« nom »)**. Comme les abscisses sont dans la liste L_1, le champ « nom » est complété par le nombre 1
- Créer une autre variable **ordo** et procéder de la même manière avec la liste 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 L_1 et L_2. Cependant il est bien de conserver les bonnes habitudes apprises lorsque nous ne faisons pas appel au module ti_system qui nécessitent cette création préalable.

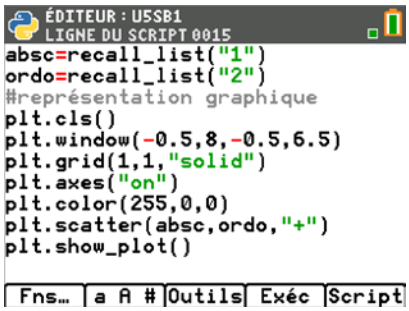
- Exécuter le script puis vérifier le contenu de vos variables **absc** et **ordo** en appuyant sur la touche var.
- Rappeler ensuite le nom de la « liste » **absc** puis valider entrer.
- Procéder de la même façon avec la liste des ordonnées (**ordo**)



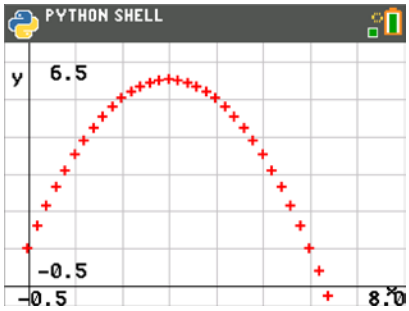
Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

c) Représentation graphique

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



Exécuter votre script (touche **F4**).



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.

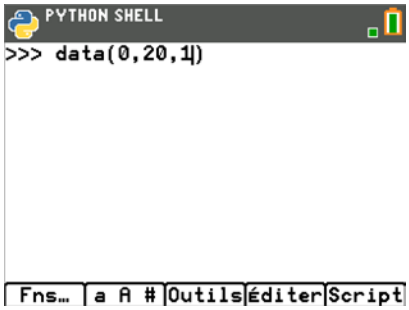
- Créer une fonction nommée **data(a,b,n)**.
- Vous allez créer deux listes de données à représenter sous forme d'un nuage de points comportant des valeurs dans un intervalle **[a ; b],** calculées avec un pas **n**.
- Dans la liste **y**, on calcule la racine carrée des valeurs de la liste **x**.
-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

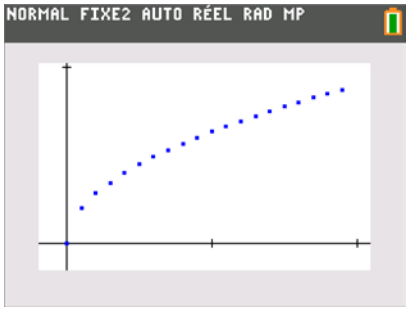
- Pour l'export de listes vers celles de la calculatrice, la syntaxe est la suivante : **store_list(« n° de liste »,nom)**. Le numéro de liste dans la calculatrice correspondant aux listes L_1 à L_6

Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/

Remarque : Attention à l'indentation, les instructions **store_list** n'ont pas à être dans la boucle. Utiliser le menu Outils (F3), puis l'option **2 : Indent <-** afin de supprimer un niveau de décalage, (Il est aussi possible d'utiliser la touche suppr.



- Exécuter votre script. On prend ici 20 valeurs de 0 à 20 par pas de 1
- Sortir de l'environnement Python et afficher la représentation graphique de vos listes L_1 et L_2.



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** & **TI-Rover**.

Ce document est mis à disposition sous licence Creative Commons
http://creativecommons.org/licenses/by-nc-sa/2.0/fr/