# Master Sciences, Technologies, Santé
## Mention Mathématiques, spécialité Enseignement des mathématiques

# Algorithmique et graphes, thèmes du second degré

# Séance de travaux pratiques n° 1

***Cette séance a pour but de maîtriser le logiciel Algobox pour ce qui concerne l'algorithmique de base, la manipulation de listes et les aspects graphiques.***

## Algorithmique de base

### Exercice 1. Décomposition d'un montant en euros

Écrire un algorithme permettant de décomposer un montant entré au clavier en billets de 20, 10, 5 euros et pièces de 2, 1 euros, de façon à minimiser le nombre de billets et de pièces.

### Exercice 2. Calcul de la $n^{ième}$ valeur d'une suite

Écrire un algorithme permettant de calculer la $n^{ième}$ valeur d'une suite de la forme $u_n = au_{n-1} + b$, $u_0 = c$ ($a$, $b$ et $c$ sont des entiers naturels entrés au clavier).

### Exercice 3. Nombres parfaits

Un nombre est parfait s'il est égal à la somme de ses diviseurs stricts (différents de lui-même). Ainsi par exemple, l'entier 6 est parfait car 6 = 1 + 2 + 3. Écrire un algorithme permettant de déterminer si un entier naturel est un nombre parfait.

## Manipulation de listes

*Quelques rappels :*

- *Les listes AlgoBox sont des listes de nombres, dont chaque élément est repéré par son rang au sein de la liste (le premier élément ayant pour rang 0).*
- *Algobox n'offre pas de fonction permettant de connaître le nombre d'éléments d'une liste et il est donc nécessaire de gérer soi-même une variable dédiée. De même, il n'y a pas d'opération prédéfinie de concaténation de listes. (Ainsi, les « listes » AlgoBox correspondent à la notion de tableau en algorithmique.)*
- *Si l'on affiche un élément de la liste dont la valeur n'a pas été initialisée, on obtient le message `Erreur`. Si l'on effectue un calcul portant sur un élément de la liste dont la valeur n'a pas été initialisée, l'exécution est interrompue avec le message :*

```
***Algorithme interrompu ligne XXX : erreur de calcul***
```

### Exercice 4. La liste est-elle monotone ?

Écrire un algorithme permettant de déterminer si une liste est ou non triée par ordre croissant ou décroissant au sens large. On commencera naturellement par saisir une liste entrée au clavier (on demandera au préalable le nombre d'éléments de cette liste), mais sans vérifier cette propriété au fur et à mesure de la saisie…

**Exercice 5. Tri par insertion**

Écrire un algorithme permettant de saisir une liste au clavier (on demandera au préalable le nombre d'éléments de cette liste), en la triant par insertion au fur et à mesure, et de l'afficher une fois la saisie terminée. Ainsi, chaque nouvel élément devra être inséré en bonne position dans la liste en cours de construction.

## Primitives graphiques

*Quelques rappels :*

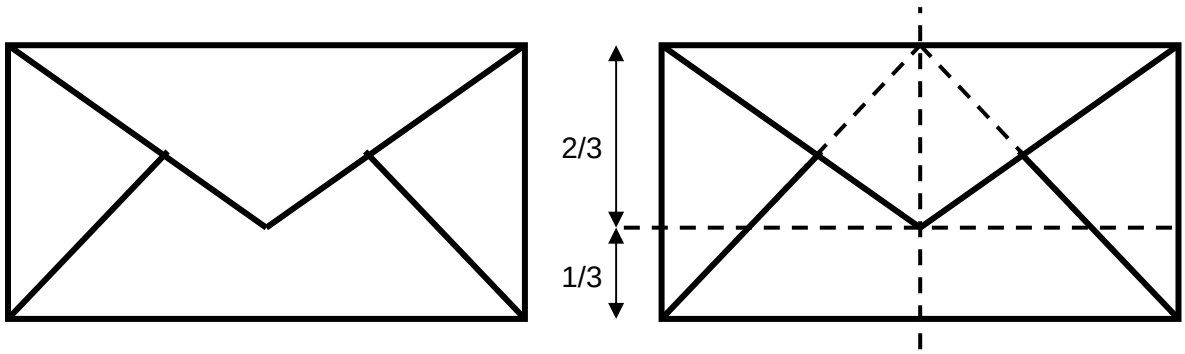
- *L'onglet `Dessiner dans un repère` permet d'accéder aux fonctions permettant de dessiner. Pour cela, il est nécessaire dans un premier temps de cocher la case `Utiliser un repère`, puis de définir les paramètres de la zone de dessin : `XMin`, `XMax` et `Graduations X`, puis `YMin`, `YMax` et `Graduations Y`.*
- *On peut ensuite utiliser les deux opérations de dessin proposées, `TRACERPOINT` (on indique les coordonnées du point) et `TRACERSEGMENT` (on indique les coordonnées du point de départ et celles du point d'arrivée). Dans chaque cas, on peut également choisir la couleur de dessin utilisée.*

**Exercice 6. Dessin de fonction**

Écrire un algorithme permettant de dessiner la courbe de la fonction $f(x) = x^2/10$, définie sur l'intervalle [-10,10]. (L'utilisateur choisira une valeur de pas et la fonction sera dessinée point par point.)

**Exercice 7. Dessin d'une enveloppe**

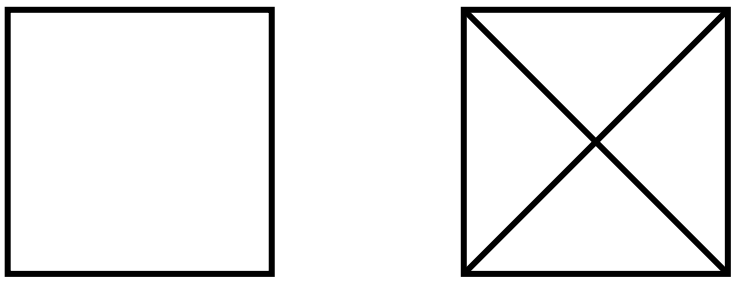
Écrire un algorithme permettant de dessiner une enveloppe selon le profil suivant (la hauteur et la largeur seront données par l'utilisateur) :




**Exercice 8. Dessin de polygones**

Écrire un algorithme permettant de dessiner un polygone régulier à N côtés, de rayon donné, en dessinant, ou pas, les « rayons », selon le modèle suivant :

Polygone à quatre côtés



sans les ravons avec les ravons

## Pour ceux qui progresseraient (trop ?) rapidement…

### Exercice 9. Fusion de deux listes triées

Écrire un algorithme permettant, à partir de deux listes triées, de construire « l'union » triée de ces deux listes. À partir des listes [3, 6, 9] et [1, 6, 8, 12, 15], on obtiendra la liste [1, 3, 6, 6, 8, 9, 12, 15]. On supposera que l'utilisateur entre correctement les deux listes triées (ou bien réutiliser l'algorithme de tri par insertion réalisé précédemment pour créer les deux listes)…

### Exercice 10. Suppression des doublons

Écrire un algorithme permettant de supprimer les doublons (éléments déjà présents) dans une liste triée donnée. À partir de la liste [3, 3, 6, 9, 9, 9, 9, 11], on obtiendra la liste [3, 6, 9, 11].

## Pour ceux qui progresseraient (vraiment) trop rapidement…

### Exercice 11. Programmation Python

Traduire en Python les algorithmes précédents…