

# DEUXIÈME SÉANCE D'ALGORITHMIQUE : LES SUITES DE FIGURES

*Proposition de scénario*

---

- Durée de l'activité : 1h00-1h30.
- Objectif : Introduire la notion de fonction. Motiver la notion de boucle.
- Déroulement : Ce TP fait suite au précédent sur les figures téléphonées.  
Après s'être approprié la notion de fonction, les élèves vont devoir écrire des fonctions puis les utiliser pour construire des suites de figures. Le TP se termine par une question qui légitimera l'introduction des boucles.
  
- Quelques remarques :
  - L'enregistreur de programme de la Tortue Xcas facilite l'écriture des fonctions puisqu'un élève peut copier/coller la suite de consignes permettant la construction d'un carré.
  - L'assistant d'Xcas est une aide précieuse pour l'écriture des premières fonctions.
  - Si la notion de polygone régulier n'a pas été vue en amont, il faudra trouver un moment pour en parler pendant le TP.



- Recopie le programme suivant :

```

Prog Edit Ajouter
fonction G()
avance;
tourne_gauche;
ffonction;;

```

- Clique sur Ok.

**b) Avec la Casio Graph 90+e**

- Ouvrir l'éditeur si nécessaire (menu F6 Fich,Cfg puis 1 Nouveau script),
- Placer le curseur avant la commande `efface;`, taper sur les touches shift et PRGM, sélectionner `fonction def` pour insérer le code définissant une fonction.
- Remplacer `f(x)` par `G()` (pour saisir la lettre G taper sur la touche alpha rouge puis la touche de label rouge G).
- En utilisant le menu tortue (raccourci clavier shift-QUIT) et les touches ALPHA et F1 (pour ;), insérer entre les deux lignes `local j;` et `return ;`<sup>1</sup> les commandes `avance;`  
`tourne_gauche();`
- Déplacer le curseur après `efface;` et insérer `G()`; . L'écran doit ressembler à :

```

x Xcas test.py *
1 fonction G()
2 local j;
3 avance;
4 tourne_gauche;
5 return;
6 ffonction;;
7 efface;
8 G();

```

- Tester (touches F6 1 ou shift-Ans)

**c) Avec Xcas pour Firefox**

- Cliquer dans la zone de saisie contenant `/* fonctions du script */`. (Rappel : cette zone apparaît quand on clique sur le bouton Prog puis Tortue puis nouveau).
- Cliquer sur le bouton `fonct`. Il apparaît alors un assistant **Nouvelle fonction ...**. Mettre `G` comme nom de la fonction et enlever `x` de la liste des arguments qui doit être vide.

**Nouvelle fonction**

nom de la fonction

liste des arguments

variables locales

valeur renvoyée

Effacer Exemple 1 Exemple 2

Validez avec le bouton vert en-dessous de valeur renvoyée.

- Insérer entre les lignes `fonction G();` et `ffonction;` les commandes `avance ;` et `tourne_gauche ;` en cliquant sur les boutons `av` et `tg`.

1. ces 2 lignes ne servent à rien pour le moment

```

1 /* fonctions du script */
Historique fonction G()
3   avance ;
4   tourne_gauche ;
5
6 ffonction;;
7

```

Historique

↑ ↑↑  
↓ ↓↓

✓ ? ← → ↑ ↓ →| ☒ ↶ ↷

find ⇒ ← rep go ✖

Validez avec le bouton vert en-dessous de la zone de saisie.

- Cliquez dans la deuxième zone de saisie commençant par la commande `efface;`, tapez la commande `G();` puis validez.

```

/* fonctions du script */
fonction G()
  avance ;
  tourne_gauche ;
ffonction;;

```

↑ ↑↑  
↓ ↓↓

Done

```

1 /* Script de déplacement*/
2 efface;
3 G();

```

↑ ↑↑  
↓ ↓↓

✓ ? ← → ↑ ↓ →| ☒ ↶ ↷

find ⇒ ← rep go ✖

△

- 2) Écris un programme "D()", qui permet d'avancer puis de tourner à droite  
On peut alors demander à Xcas d'exécuter une des suites de commandes suivante :

```

efface;
G();
D();
G();
D();
G();
G();
avance(20);
tourne_gauche;
D();
G();
avance(20);
tourne_gauche;

```

```

efface;
repete(2,G(),D());
repete(2,G());
avance(20);
tourne_gauche;
D();
G();
avance(20);
tourne_gauche;

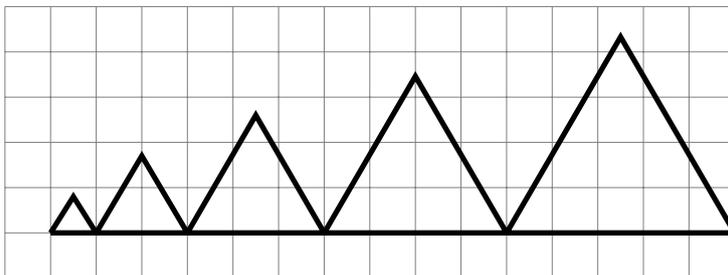
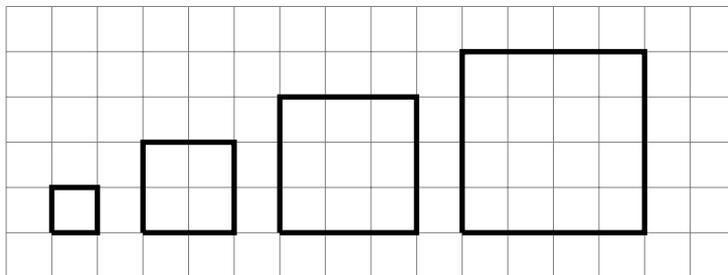
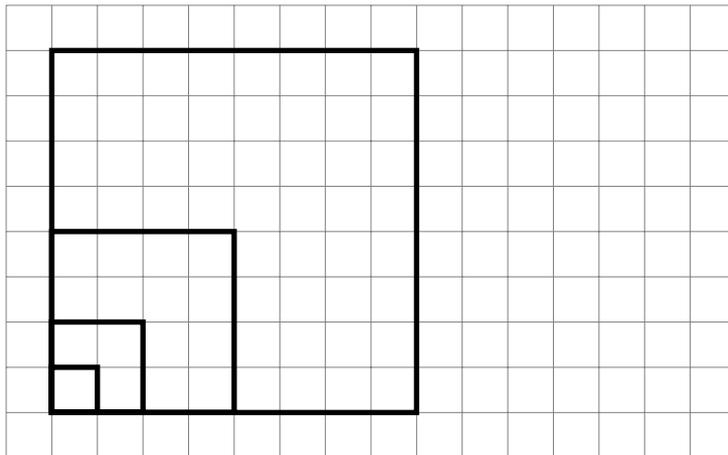
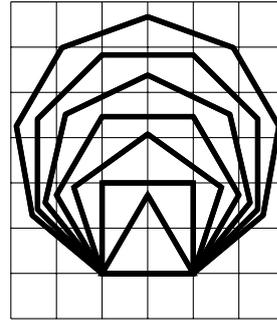
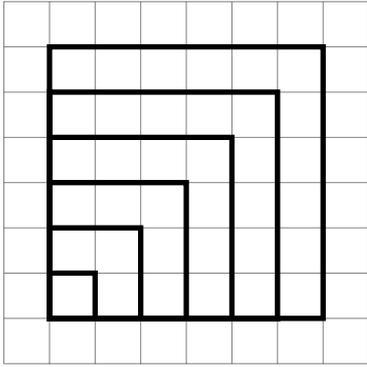
```

Que se passera-t-il ?

- Écris un programme qui s'appelle "Carré()" et qui construit un carré de côté 1 unité.
- Écris un programme qui s'appelle "Carré(L)", qui prend comme argument "L" et qui construit un carré de côté L unités lorsqu'il est exécuté.
- Écris un programme qui s'appelle "Hexagone()" et qui construit un hexagone régulier de côté 1 unité.
- Écris un programme qui s'appelle "Polygone(n,L)", qui prend comme argument "n" et "L" et qui construit un polygone régulier de n côtés de longueur L unités lorsqu'il est exécuté.

## ■ DES SUITES DE FIGURES

Construis les figures ci-dessous :



*Nom :*

*Prénom :*

*Classe :*

## COMPTE-RENDU

---

---

1) Recopie tes fonctions "Carré()", "Carré(L)", "Hexagone()" et "Polygone(n,L)".

2) a) Explique comment tu as procédé pour construire les suites de figures.

b) Utiliserais-tu la même méthode pour construire une suite de 30 figures ou en privilégierais-tu une autre? Explique.