

Introduction à l'informatique

Exercices — Série 5

Introduction

Cette série d'exercices porte sur l'écriture de programmes C graphiques. Pour ce faire, une bibliothèque de fonctions permettant de générer des images au format SVG est mise à votre disposition. Cette bibliothèque est implémentée sous la forme de deux fichiers `draw.h` et `draw.c` qui peuvent être téléchargés depuis la page WWW du cours. Cette bibliothèque fournit trois fonctions :

- `draw_init(file_name, largeur, hauteur)` initialise un nouveau fichier au format SVG appelé `file_name`, destiné à contenir une image de dimension (`largeur`, `hauteur`). Par exemple, on écrira `draw_init("mon_fichier.svg", 800, 600);` pour créer une image de 800×600 pixels placée dans le fichier `mon_fichier.svg` du répertoire courant. Cette fonction doit être appelée préalablement à toute autre opération de la bibliothèque.
- `draw_line(x1, y1, x2, y2, trait, r, g, b)` trace une ligne entre les points de coordonnées (`x1`, `y1`) et (`x2`, `y2`), d'épaisseur `trait` et de couleur (`r`, `g`, `b`). Les coordonnées `x1`, `y1`, `x2`, `y2` ainsi que la valeur de `trait` sont exprimées en pixels. L'origine du système d'axes est situé en haut et à gauche de l'image; l'axe horizontal va de gauche à droite et l'axe vertical de haut en bas. Une couleur (`r`, `g`, `b`) est caractérisée par ses composantes respectivement rouge (`r`), verte (`g`) et bleue (`b`), comprises entre 0 et 255.
- `draw_close()` finalise le fichier courant. Cette fonction doit toujours être appelée, et doit l'être en tant que dernière opération de la bibliothèque.

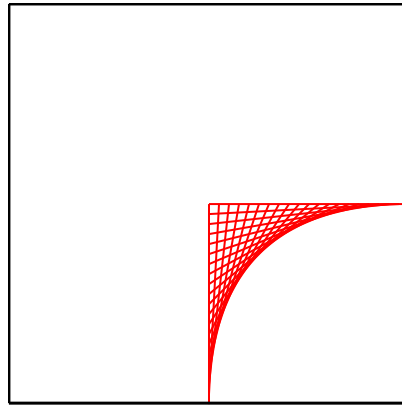
Chacune de ces fonctions retourne 0 en cas de succès, et une valeur négative en cas d'erreur. Pour certains exercices, il sera également nécessaire d'utiliser la bibliothèque mathématique :

- Le fichier d'en-tête standard à inclure est `math.h`.
- Dans le cas d'une compilation en ligne de commande à l'aide de `gcc`, l'option "`-lm`" doit être ajoutée à celles du compilateur.
- Les fonctions `double sin(double)` et `double cos(double)` correspondent aux fonctions trigonométriques standard.
- La constante `M_PI` fournit la valeur approximée de π .

Enfin, signalons que les fichiers graphiques au format **SVG** peuvent être visualisés par la plupart des programmes permettant d'afficher des images, ainsi que par les navigateurs WWW modernes.

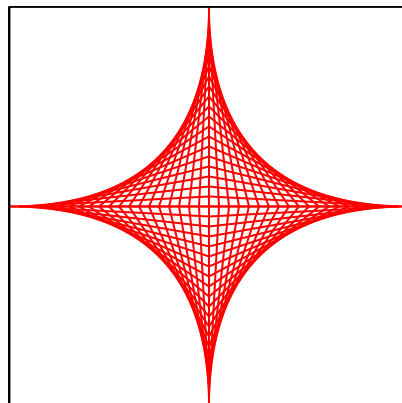
Exercice 1

1. La première étape de cet exercice consiste à écrire un programme C qui crée une image de 400×400 pixels, et y simule le mouvement d'une latte rigide de 200 pixels de longueur, dont les extrémités se déplacent sur deux segments horizontal et vertical issus du centre de l'image. L'objectif est d'arriver à une image similaire à celle-ci :



Notes : Pour cette image, 21 positions de la latte sont simulées. L'épaisseur du trait est de 2 pixels.

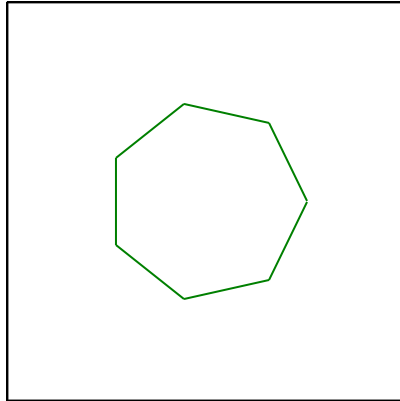
2. L'objectif est à présent de compléter ce programme de façon à arriver à une image similaire à celle-ci, en faisant à présent se déplacer la latte sur deux droites horizontale et verticale se croisant au centre de l'image :



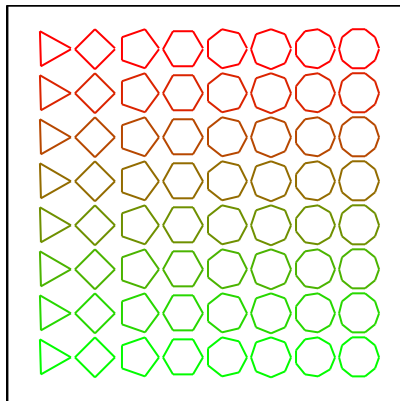
Exercice 2

1. La première étape de ce problème consiste à écrire une fonction `C` capable de dessiner un polygone régulier. Le nombre de côtés de ce polygone, les coordonnées de son centre, le rayon de son cercle circonscrit, l'épaisseur du trait et la couleur de celui-ci doivent être des paramètres de cette fonction.

Par exemple, un appel à cette fonction pour un nombre de côtés égal à 7, un rayon de 100 pixels, et un polygone centré sur le milieu d'une image de 400×400 pixels devrait produire un résultat similaire à celui-ci :



2. La deuxième étape consiste à écrire un programme `C`, exploitant la fonction obtenue au point 1, capable de générer une image similaire à celle-ci :



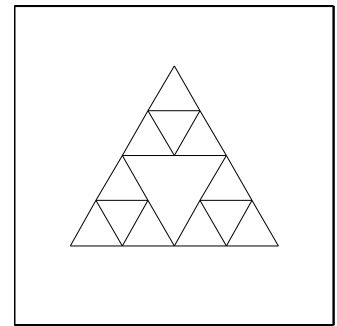
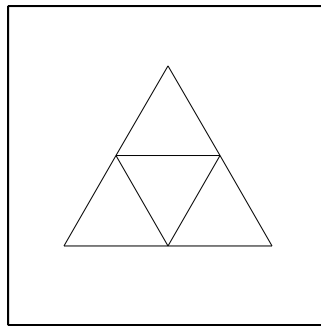
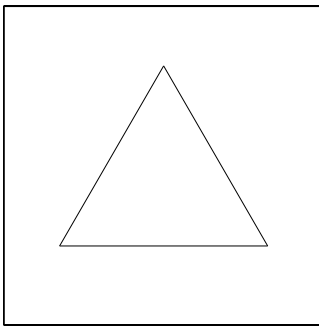
(Dans cette image, le nombre de côtés des polygones varie de 3 à 10 lorsqu'on se déplace de gauche à droite, et leur couleur varie du rouge au vert lorsqu'on se déplace de haut en bas.)

Exercice 3

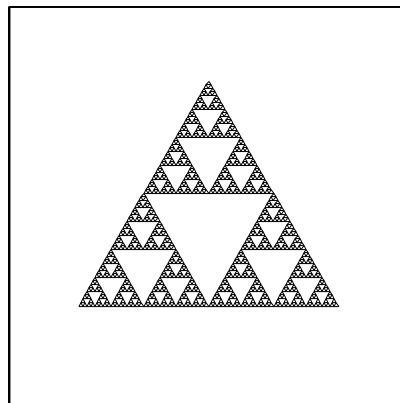
Le but de cet exercice est d'écrire un programme C capable de générer une image de 400×400 pixels approximant le *triangle de Sierpinski*. Ce triangle se construit grâce à la procédure suivante :

1. On dessine un triangle équilatéral.
2. On découpe ce triangle en quatre triangles plus petits en joignant les milieux de ses côtés.
3. On répète la même procédure dans chacun de ces petits triangles, à l'exception de celui situé au centre.

Par exemple, les trois premières applications de cette procédure produisent les images suivantes :



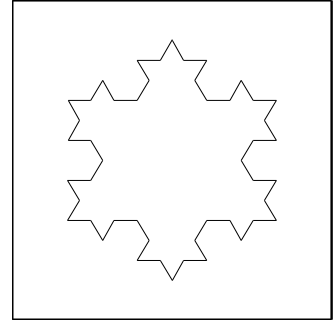
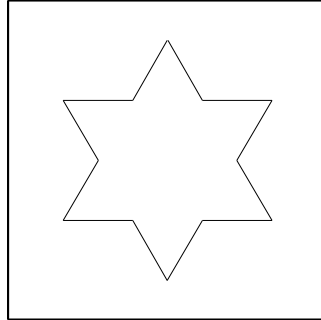
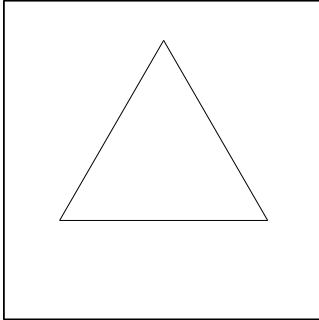
Le résultat obtenu après 6 applications est le suivant :



Exercice 4

Cet exercice porte sur l'écriture d'un programme C capable de dessiner une approximation du *Flocon de Koch*. Celui-ci s'obtient en appliquant plusieurs fois de suite une transformation particulière à un triangle équilatéral. Cette transformation consiste à remplacer chaque segment de droite de la figure par une ligne brisée, produite en découpant le segment en trois parties égales, en construisant un triangle équilatéral extérieur à la figure sur la partie centrale, et en retirant finalement cette partie centrale.

Les trois premières étapes de cette procédure produisent les résultats suivants :



L'image obtenue après 6 étapes est la suivante :

