

## TP n° 3

**I3 : Tortue**

La tortue est une méthode de programmation graphique. On utilise un curser (la tortue à proprement parler) que l'on déplace de deux façons :

- En ligne droite, en donnant la longueur du chemin à parcourir ;
- En effectuant une rotation, en donnant l'angle.

**I) Utilisation de la bibliothèque libTortue**

On fait pour cela appel à une bibliothèque dynamique externe : libTortue.so. Elle se trouve sur les machines des salles TP dans le répertoire `/usr/lib`, qui est normalement dans le chemin utilisé par l'éditeur de liens. Les prototypes de ses fonctions sont déclarés dans le fichier `Tortue.h`, qui se trouve dans le répertoire `/usr/include`. Vous pouvez le regarder pour voir les fonctions qui sont à votre disposition.

Lorsque vous écrivez un programme utilisant des fonctions de cette bibliothèque, vous devez donc inclure le fichier contenant la déclaration des prototypes de ses fonctions au début de votre programme :

```
1 #include <Tortue.h>
```

Vous devez également indiquer à l'éditeur de liens qu'il faut utiliser la bibliothèque libTortue. Votre ligne de compilation devient donc, par exemple :

```
1 $ gcc -o carre carre.c -lTortue
```

**II) Tracer avec la tortue****a) Initialisation et finalisation**

On ouvre la fenêtre graphique avec la fonction `ouvre()` (sans arguments), on la ferme avec la fonction `ferme()` (sans arguments). Tous les déplacements de la tortue doivent impérativement être effectués après l'ouverture et avant la fermeture de la fenêtre graphique.

**b) Déplacements**

On déplace la tortue en ligne droite avec la fonction `avance( longueur )`, si `longueur` est la longueur du déplacement en nombre de pixels.

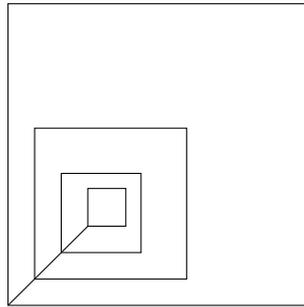
On la fait tourner sur elle-même dans le sens horaire avec la fonction `tourne( angle )`, si `angle` est l'angle de la rotation en degrés.

**Exercice 1** [Dessiner un carré]

Écrivez un programme C qui demande à l'utilisateur de saisir un entier et dessine un carré de côté égal à cet entier en utilisant la tortue. Vous devrez utiliser une boucle. Indice : un carré a quatre côtés de longueurs égales et quatre angles droits.

**Exercice 2** [Succession de carrés]

1. Reprenez le code de l'exercice précédent et écrivez une procédure qui trace un carré dont la longueur du côté est passée en argument de la fonction.



- Écrivez un programme en C qui demande à l'utilisateur un nombre de carrés et une longueur de côté minimale et réalise le dessin ci-dessus.

Le segment qui relie chaque carré est de longueur fixe et égale au côté du plus petit carré et forme un morceau de diagonale de chaque carré. La longueur du côté de chaque carré autre que le plus petit est égale au double de la longueur du côté du carré juste au-dessous.

### Exercice 3 [Cercle]

- La tortue ne propose pas de fonction permettant de tracer un cercle. Écrivez une procédure `cercle( int nbCotes, int lgCotes )` en C qui approxime un cercle comme un polygone régulier à  $nbCotes$  côtés de longueur  $lgCotes$  et d'angle  $360/nbCotes$ .
- Modifiez votre fonction `cercle()` pour lui faire prendre un paramètre en plus : la proportion de cercle à tracer (1/2, 1/4...), passée sous la forme d'un nombre décimal à virgule flottante.
- Modifiez votre fonction `cercle()` pour lui passer non plus la longueur des côtés, mais le rayon du cercle. Attention, il y a un tout petit calcul de trigonométrie pour trouver la relation entre le rayon du cercle et la longueur du côté.

Vous pourrez utiliser la fonction `sin()` de la bibliothèque de mathématiques (en-tête : `math.h`, bibliothèque : `libm`) et la constante  $\pi$  `M_PI` (également définie dans `math.h`).

Pour vérifier que votre cercle est correct, vous pourrez tracer un demi-cercle et son diamètre.

### Exercice 4 [Jouons à la marelle]

En utilisant la procédure `carre()` et la procédure `cercle()` des exercices précédents, dessinez une marelle.

