

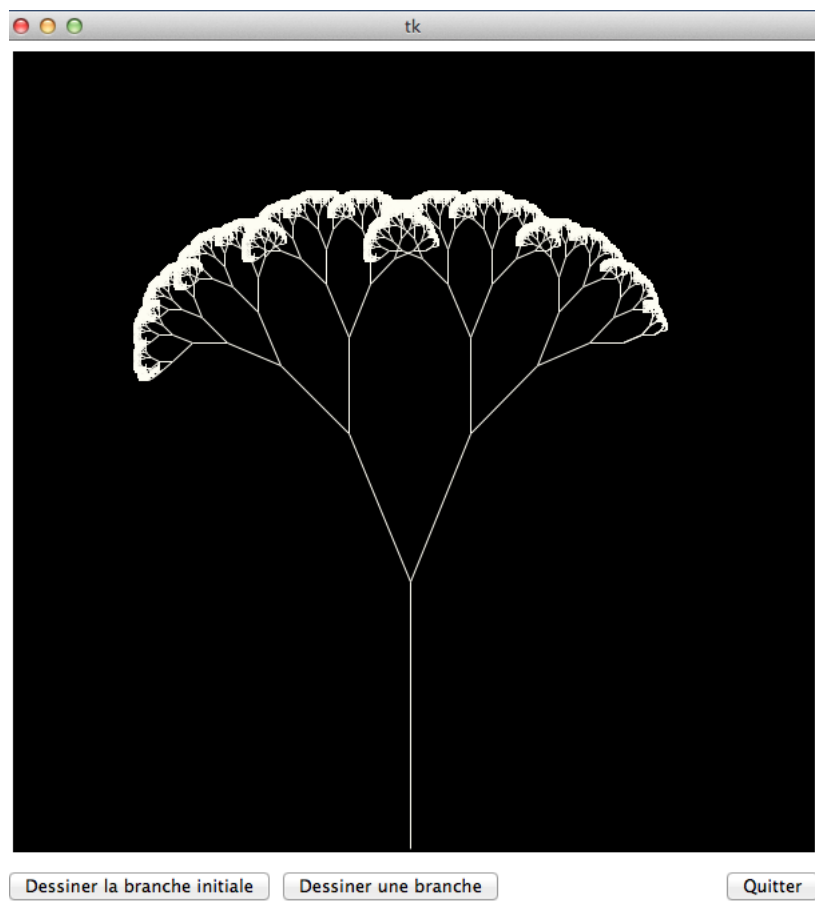
TCSS5AC - Info I - Examen

Durée : 2h

Le 13 Décembre 2018

Tout document autorisé. Envoyez votre code à Cecile.Pierrot@inria.fr avant 16h. Tout mail reçu en retard sera pénalisé d'un point négatif par minute de retard. N'oubliez pas votre nom en tête de document, en commentaire.

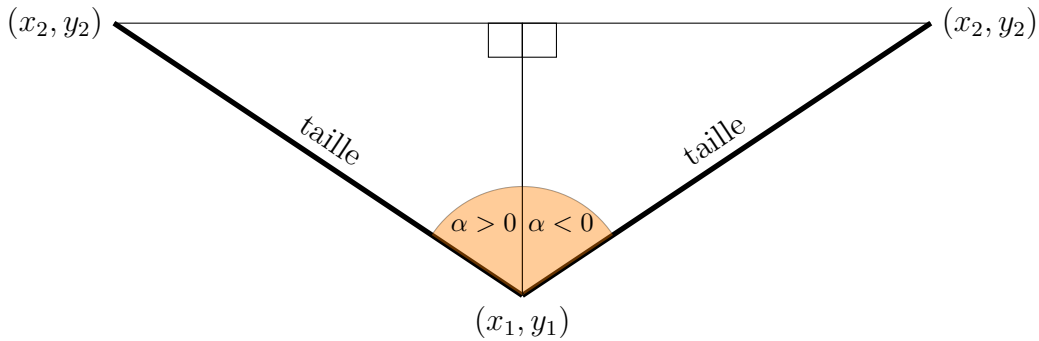
Exercice 1 : Dessin d'un arbre en fractale (16 points) Le but de cet exercice est de dessiner un arbre, par exemple comme celui-ci :



1. Affichez une fenêtre qui comporte un canvas de 600 pixels de côté, de fond noir, et qui comporte un bouton "Quitter" permettant de fermer la fenêtre, et deux boutons

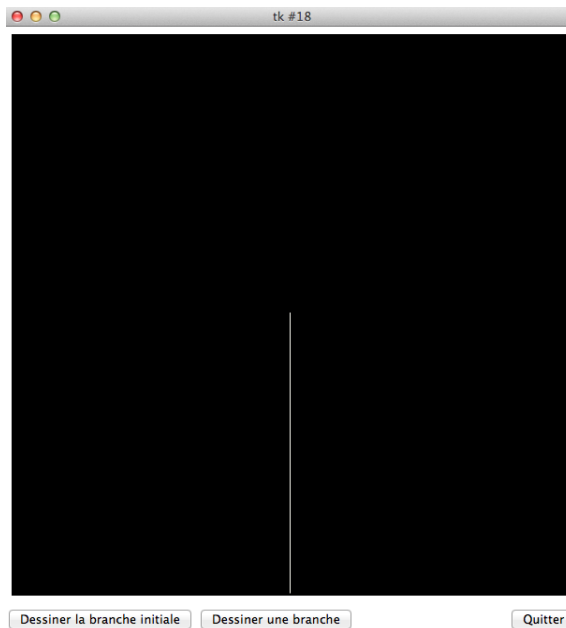
”Dessiner une branche” et ”Dessiner la branche initiale” qui pour l’instant ne font rien.

2. Ecrivez une fonction *tracerbranche* qui étant donné les coordonnées x_1 et y_1 d’un point A , une longueur *taille*, un angle α , et une couleur *couleur*, calcule les coordonnées du point B de coordonnées (x_2, y_2) d’arrivée de la branche, et trace cette branche de la bonne couleur dans le canvas. On veut de plus que la fonction renvoie le couple (x_2, y_2) calculé. L’angle α doit donner la direction de tracé de la branche, par rapport à l’axe verticale. Si α est positif alors on dessinera vers la gauche, dans le cas contraire on dessinera vers la droite.

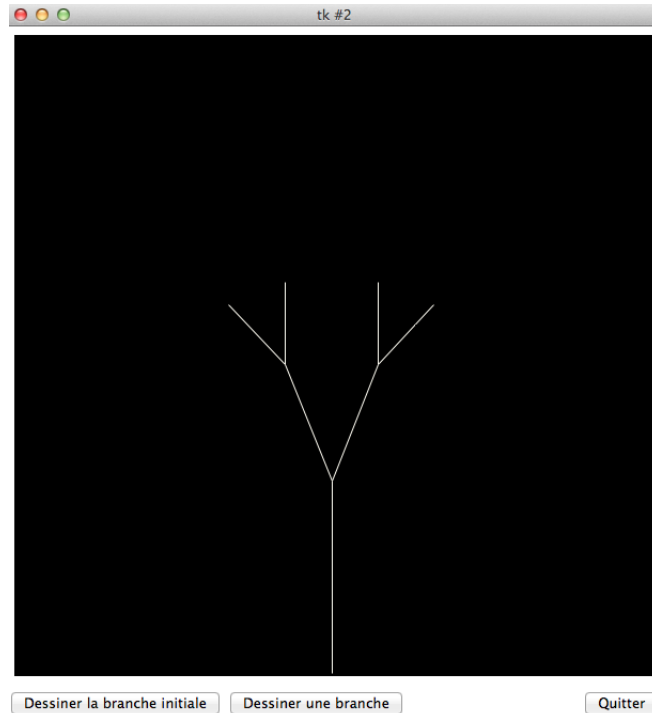


Vous pouvez vous aidez de la figure ci-dessus, et des fonctions *cosinus* et *sinus* de la bibliothèque *math*. Attention :

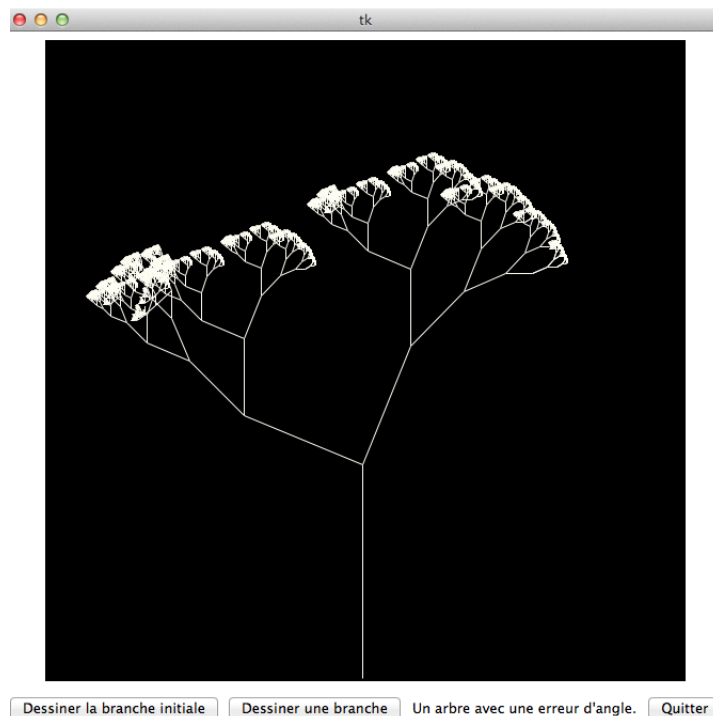
- ces fonctions travaillent sur des angles en radian.
 - il peut y avoir des soucis aux cas extrêmes, π , $\pi/2$ etc... Vous les verrez apparaitre plus loin.
3. Modifiez votre code de telle sorte qu’un clic sur le bouton ”Dessiner la branche initiale” donne le résultat suivant :



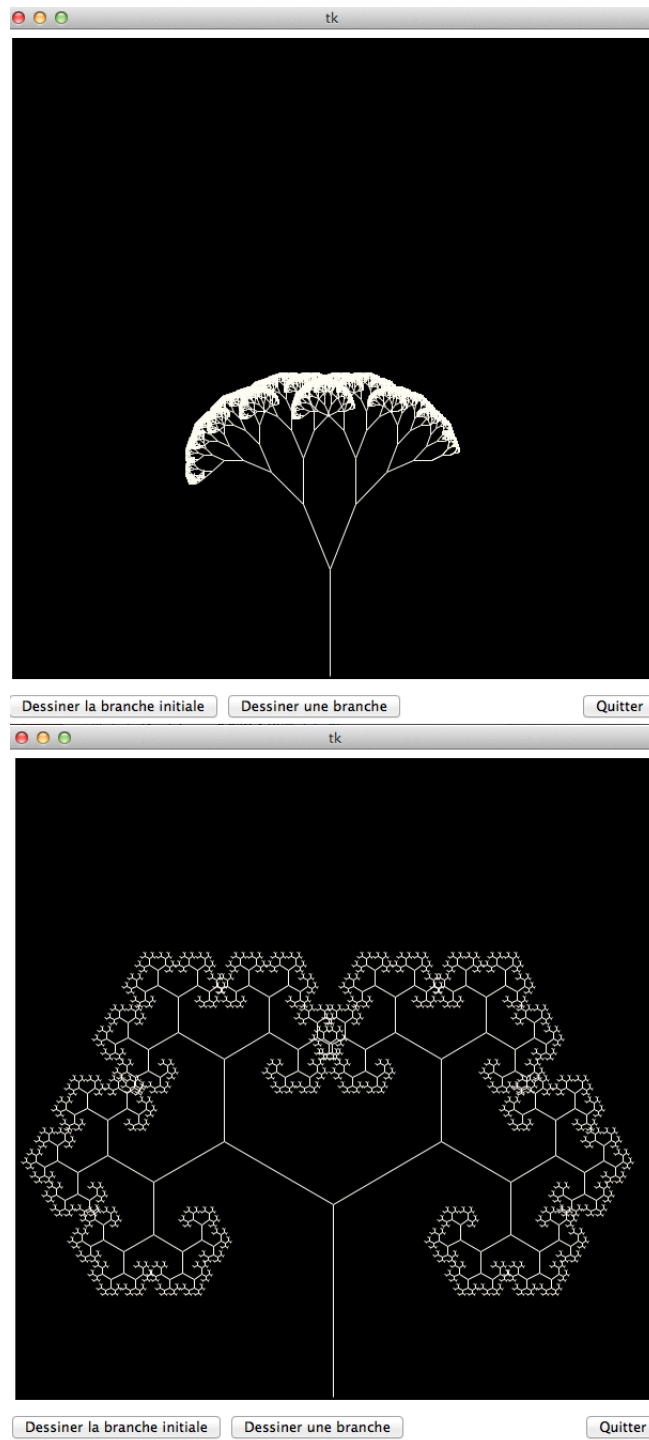
4. Modifiez votre code de telle sorte qu'un clic sur le bouton "Dessiner une branche" dessine la branche initiale, que deux clics dessinent les deux branches suivantes etc. Chaque clic doit donner le double de branches par rapport à l'étape précédente. La longueur des branches diminue d'un facteur constant (par exemple 0,65) à chaque étape. Les nouvelles branches partent des branches précédentes. Après 3 clics de ce bouton vous devez par exemple obtenir l'image suivante :



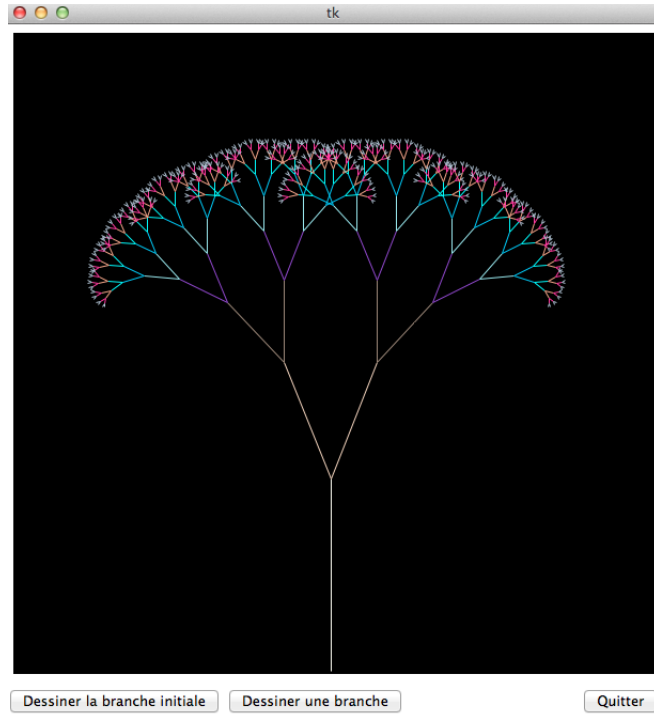
Un exemple d'arbre qui a un problème d'angle :



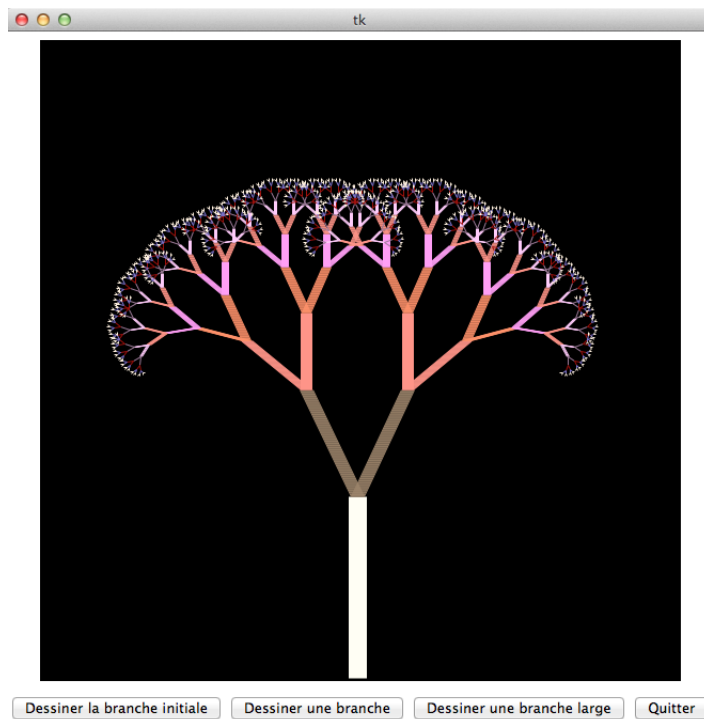
Et deux exemples d'arbres corrects. Les formes dépendent de vos paramètres de taille de branches et d'angles, que vous pouvez modifier pour obtenir plusieurs cas. Essayez d'obtenir les deux arbres ci-dessous, et indiquez clairement dans votre code les choix de paramètres à faire.

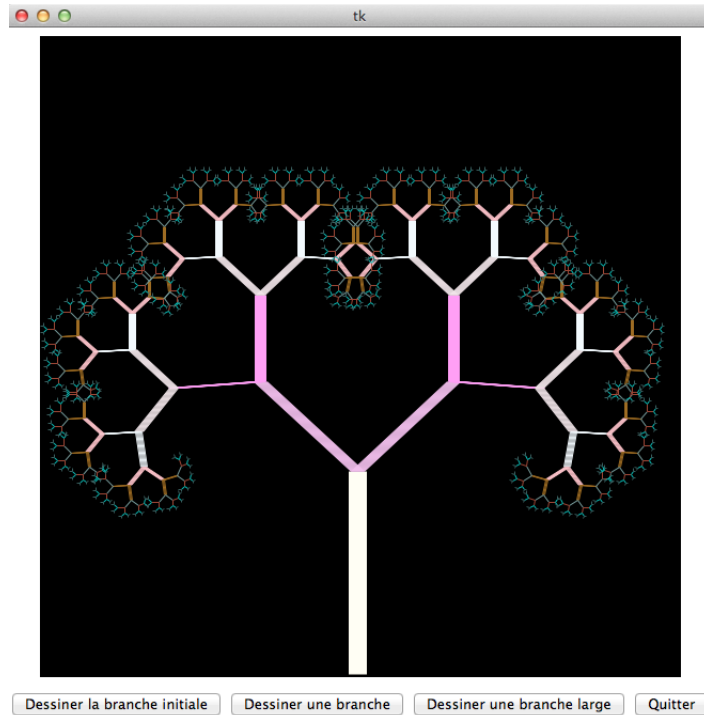


5. Question bonus. Modifiez votre code pour que chaque génération de branches change de couleur de manière aléatoire. On obtiendra par exemple un dessin comme celui-ci :



6. Question bonus bis. Les branches sont un peu trop fines. Modifiez votre code pour que le tronc soit bien épais, et qu'à mesure que l'on monte dans les branches de l'arbre, celles-ci deviennent de moins en moins larges. Par exemple :





Exercice 2 : Ecriture en binaire et complexité (4 points) Comme vous le savez, pour convertir un nombre entier positif N de la base décimale à la base binaire, il faut opérer par des divisions successives du nombre N par 2. Les restes des divisions constituent la représentation binaire.

1. Ecrire une fonction récursive **binaire** permettant de renvoyer la représentation binaire d'un nombre N . La signature de cette fonction est la suivante :

```
int --> int
```

Et voici un jeu de tests - à inclure dans votre code :

```
assert binaire(0)==0
assert binaire(1)==1
assert binaire(2)==10
assert binaire(13)==1101
```

2. En commentaire après votre jeu de test, donnez une estimation de la complexité de cette fonction. Conseil : estimez le nombre de parcours d'une certaine boucle, et dites pourquoi c'est ce nombre qui donne l'ordre de grandeur de la complexité de l'algorithme.