Informations techniques PC Suze :
— Pour démarrer *Processing* : clic sur `K` en haut à droit → Éducation → Processing.
  • Si le ménu `K` n'existe pas : clic droit sur le bureau → Ajouter un panneau → Tableau de bord par défaut
— Pour démarrer une *console* : clic sur `K` en haut à droit → Application pédagogiques → Terminal → Konsole.
— Pour démarrer un navigateur/explorer de fichiers : clic sur `K` → Utilitaires → Dolphin.
— Pour lancer une commande : clic droit sur le bureau → Exécuter une commande (ou Alt + Space)
— Pour démarrer une machine Windows : `K` → Machines Virtualbox → Info_Windows_XP ;
— Pour modifier un fichier, clic droit sur le fichier → Ouvrir avec KWrite (ou autre éditeur de votre choix).

ED/TP Récursivité et expressions régulières

Exercice 0 Qu'est ce qui caractérise une fonction récursive ?

Exercice 1 Écrire une fonction récursive qui calcule le $n^{\text{ème}}$ terme de la suite de Fibonnaci. Rappel : $F(n) = F(n-1) + F(n-2)$ pour tout $n > 2$ et $F(1) = F(2) = 1$.

Exercice 2 Faites tourner le programme ci-après. Modifier ce programme pour remplir avec des carrés une zone plus grande, de 600×600 pixels.

```
void rec(int a, int b, int taille){
    rect(a,b,taille,taille);
    if(taille<10)
        return;
    rec(a,b,taille/2);
    rec(a+taille/2,b+taille/2,taille/2);
}
void setup(){
};
void draw(){
    rec(0,0,100);
}
```

Exercice 3 Indiquer le résultat des commandes ci-dessous. Attention : pour tester, il faut démarrer une console : click sur `K` en haut à gauche→ Application pédagogiques → Terminal → Konsole.

```
ls /bin/z*
ls /bin/[y-z]*
ls /bin/*t

echo -e "aaa\nbbb\nccc" > fic.txt

grep a fic.txt
sed "s/a/b/g" fic.txt
sed "s/a//g" fic.txt>fic_no_a.txt
ls -l|sed "s/-/TOTO/g"
ls /bin |sed "s/z.*/TOTO/g"

ls /bin |grep ^z    #observer la différence par rapport à "grep z"
```

Exercice 4 Prenez un programme que vous avez déjà réalisé et utiliser `sed` pour changer le nom d'une variable. Sauvegarder le nouveau programme dans un ficher `test.pde` et vérifier s'il tourne correctement.

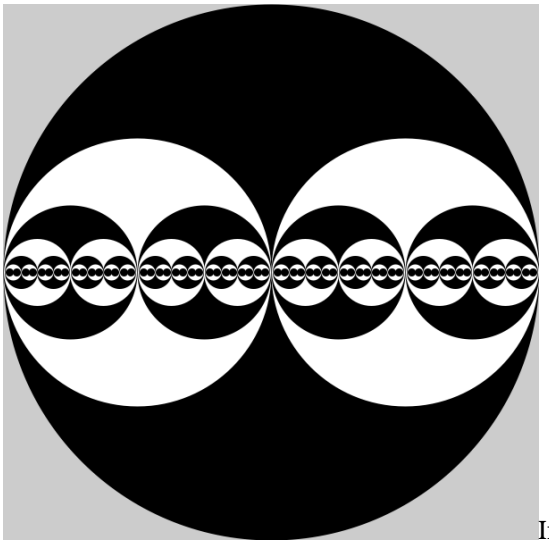
Exercice 5 Créer un fichier `fic.txt` avec le contenu suivant :

```
Anna, 10+10 = 20
Pierre, 10+9 = 19
Toto,   6+8 = 14
```

Réaliser sur ce fichier les opérations suivantes à l'aide de `cat`, `sed` ou `grep` :

1. afficher simplement le fichier dans la console
2. afficher que les lignes qui concernent `Toto`
3. remplacer `+` avec `,`.
4. remplacer `=` avec `,`.
5. remplacer `=` et `+` avec `,`. Indication : testez la commande `ls |sed "s/[ab]/c/g"`
6. afficher que les lignes avec 20 dans la dernière colonne (ctd, note finale de 20)
7. enlever la dernière colonne (les caractères à partir de `=` jusqu'à la fin de la ligne)

Exercice 6 Ecrire une fonction récursive générant une image de ce genre.



Indication : tracer dans une première phase les cercles sans couleur (en utilisant `noFill()`). La fonction devrait d'abord tracer un cercle de rayon r fourni comme argument. Ensuite, il faut faire appel récursivement à la fonction elle-même en lui passant $r/2$ comme argument.

Exercice 7 Faites tourner le programme ci-après. Il est possible de déplacer le carré rouge grâce aux touches `o` et `p`. Modifier la méthode `keyPressed` pour interdire au carré rouge de se déplacer "à travers les mûrs" : on ne doit pas autoriser un déplacement sur un carré occupé.

```
int lignes = 7;
int cols  = 5;
int[][] carte={
                {1,1,1,1,1},
                {1,0,0,1,0},
                {1,1,0,0,1},
                {1,0,0,1,1},
                {1,0,1,0,0},
                {1,0,0,0,1},
                {1,1,1,1,1}};
int iMonRectangle=3;
int jMonRectangle=2;
void drawCarte(){
    for(int i=0;i<lignes;i++)
        for(int j=0;j<cols;j++){
            if(carte[i][j]==1)
                fill(0);
            else
                fill(255);
            if(carte[i][j]==100){
                fill(0,0,255);
            }
            if((i==iMonRectangle)&&(j==jMonRectangle)){
                fill(255,0,0);
```

```
                }
                rect(j*50,i*50,50,50);
            }
    }
    void setup(){
        size(700,700);
        drawCarte();
    }
    void draw(){
    }
    void keyPressed(){
        int i = iMonRectangle;
        int j = jMonRectangle;
        if(key=='p')
            j++;
        if(key=='o')
            j--;

        if((i!=-1)&&(j!=-1)&&(i!=lignes)&&(j!=cols)){
            iMonRectangle = i;
            jMonRectangle = j;
        }
        drawCarte();
    }
```

Exercice 8 Modifier le programme pour pouvoir déplacer le rectangle rouge en haut et en bas, à l'aide des touches "a" et "q".

Exercice 9 Modifier la méthode `keyPressed()` pour ajouter la fonctionnalité suivante. Si le rectangle rouge arrive aux frontières du labyrinthe, il faut afficher un message "vous avez réussi" !