

Informatique PCSI
Corrigé suite TP 5
Exercice 1

```
def pgcd(a, b):
    if b == 0:
        return a
    else:
        return pgcd(b, a%b)
```

Exercice 2

```
def nettoie_rec(liste, k=0):
    if k < len(liste) - 1:
        if liste[k] != liste[k+1]:
            nettoie_rec(liste, k+1)
        else:
            del liste[k]
            nettoie_rec(liste, k)
```

Exercice 3

La fonction récursive prend un paramètre i prenant les valeurs de 0 à 180.

```
def dessin(i):
    if i < 180:
        color(couleurs[i%6])
        forward(i)
        right(59)
        dessin(i+1)

dessin(0)
```

Exercice 4

1. On examine les valeurs successives des variables.
 - Nous avons g et d qui valent respectivement 0 et 11. Donc m prend la valeur 5 et $liste[5]$ vaut 13. Puisque $10 < 13$, on procède à l'appel `mystere(10, liste, 0, 4)`.
 - Nous avons g et d qui valent respectivement 0 et 4. Donc m prend la valeur 2 et $liste[2]$ vaut 7. Puisque $7 < 10$, on procède à l'appel `mystere(10, liste, 3, 5)`.
 - Nous avons g et d qui valent respectivement 3 et 5. Donc m prend la valeur 4 et $liste[4]$ vaut 10. Donc la fonction renvoie l'indice 4.

2. On examine encore les valeurs successives des variables.

- Nous avons `g` et `d` qui valent respectivement 0 et 11. Donc `m` prend la valeur 5 et `liste[5]` vaut 13. Puisque $18 > 13$, on procède à l'appel `mystere(18, liste, 6, 11)`.
- Nous avons `g` et `d` qui valent respectivement 6 et 11. Donc `m` prend la valeur 8 et `liste[8]` vaut 17. Puisque $18 > 17$, on procède à l'appel `mystere(18, liste, 9, 11)`.
- Nous avons `g` et `d` qui valent respectivement 9 et 11. Donc `m` prend la valeur 10 et `liste[10]` vaut 23. Puisque $18 < 23$, on procède à l'appel `mystere(18, liste, 9, 10)`.
- Nous avons `g` et `d` qui valent respectivement 9 et 10. Donc `m` prend la valeur 9 et `liste[9]` vaut 20. Puisque $18 < 20$, et le test `g == m` est vérifié, la fonction renvoie `False`.

Exercice 5

```
def palindrome(ch):
    """ ch est de type str
        renvoie True si ch est un palindrome, False sinon """
    n = len(ch)
    if n <= 1:
        return True
    else:
        return ch[0] == ch[n-1] and palindrome(ch[1:n-1])
```