

Exercice 1 :

```
def saisir():
    nb = int(input("nombre d'étudiants à rentrer : "))
    fichier=open("concours.txt", 'a', encoding='utf8') #Ouverture du fichier concours.txt
    for i in range(nb):
        print(f"| Etudiant numéro {i+1} : |")
        ncin = int(input("Numéro de carte d'identité : "))
        if type(ncin)!=int:
            raise TypeError("Le numéro de carte d'identité doit être un entier")
        nom = input("Nom : ")
        if type(nom)!=str:
            raise TypeError("Le nom doit être une chaîne de caractère")
        prenom = input("Prenom : ")
        if type(prenom)!=str:
            raise TypeError("Le prénom doit être une chaîne de caractère")
        age = int(input("Age : "))
        decision = input("Décision (admis, refusé ou ajourné) : ")
        if type(decision)!=str:
            raise TypeError("La décision doit être une chaîne de caractère")
        if decision not in ["admis", "ajourné", "refusé"]:
            raise ValueError("La saisie de la décision est fausse")
        fichier.write(f"{ncin};{nom};{prenom};{age};{decision}\n")
    fichier.close()

def admis():
    dest = open("admis.txt", 'w', encoding='utf8')
    source = open("concours.txt", 'r', encoding='utf8')
    for line in source: #Parcours des lignes du fichier source et ajoute dans le fichier source quand la
        décision est admis
        splitted_line = line.strip().split(";")
        if splitted_line[4] == "admis":
            dest.write(f"{splitted_line[0]};{splitted_line[1]};{splitted_line[2]};{splitted_line[3]};{splitted_line[4]}\n")
    dest.close()
    source.close()

def attente():
    dest = open("attente.txt", 'w', encoding='utf8')
    source = open("admis.txt", 'r', encoding='utf8')
    for line in source:
        splitted_line = line.strip().split(";")
        if int(splitted_line[3]) > 30:
            dest.write(f"{splitted_line[0]};{splitted_line[1]};{splitted_line[2]}\n")

import matplotlib.pyplot as plt

def statistique(dec):
    if type(dec) != str:
        raise TypeError("La décision est une chaîne de caractère")
    if dec not in ["admis", "ajourné", "refusé"]:
        raise ValueError("La saisie de la décision est fausse")
```

```

nb_admis = 0
nb_refus = 0
nb_ajour = 0
nb_total = 0
fichier = open("concours.txt", 'r', encoding='utf8')
for line in fichier:
    nb_total += 1
    splitted_line = line.strip().split(";")
    if splitted_line[4] == "admis":
        nb_admis += 1
    elif splitted_line[4] == "refusé":
        nb_refus += 1
    elif splitted_line[4] == "ajourné":
        nb_ajour += 1
fichier.close()

if dec == 'admis':
    print(100*nb_admis/nb_total, "%")
if dec == 'ajourné':
    print(100*nb_ajour/nb_total, "%")
if dec == 'refusé':
    print(100*nb_refus/nb_total, "%")
plt.figure(figsize=(8, 8))
x = [nb_admis, nb_refus, nb_ajour]
plt.pie(x, labels=['Admis', 'Refusés', 'Ajournés'], colors=['red', 'blue', 'green'], autopct=lambda x:
str(round(x, 2)) + '%')
plt.legend()
plt.show()

def supprimer():
    f = open("admis.txt", 'r')
    lines = f.readlines()
    f.close()
    f = open("admis.txt", 'w')
    for line in lines:
        splitted_line = line.strip().split(";")
        if int(splitted_line[3]) <= 30 :
            f.write(line)
    f.close()

def mainExo1():
    print("----- Début du programme -----")
    print("| Test de la fonction saisie |")
    try :
        saisir()
    except ValueError as erreur :
        print(erreur)
    except TypeError as erreur :
        print(erreur)
    print("| Affichage |")
    with open("concours.txt", 'r', encoding='utf8') as file:
        for line in file:
            print(line)

    print("| Test de la fonction admis |")
    admis()
    print("| Affichage |")

```

```

with open("admis.txt", 'r', encoding='utf8') as file:
    for line in file:
        print(line)

print("| Test de la fonction attente |")
attente()
print("| Affichage |")
with open("attente.txt", 'r', encoding='utf8') as file:
    for line in file:
        print(line)

print("| Test de la fonction statistique pour la décision ajourné |")
try :
    statistique("ajourné")
except ValueError as erreur :
    print(erreur)
except TypeError as erreur :
    print(erreur)

print(" | Test de la fonction supprimer |")
supprimer()
print("Affichage du fichier admis.txt après suppression : ")
with open("admis.txt", 'r', encoding='utf8') as file:
    for line in file:
        print(line)

```

Essais :

```
| Test de la fonction saisie |
nombre de candidats à rentrer : 7
| Candidat numéro 1 : |
Numéro de carte d'identité : 273
Nom : LOPES
Prenom : ISABELLE
Age : 23
Décision (admis, refusé ou ajourné) : admis
| Candidat numéro 2 : |
Numéro de carte d'identité : 734
Nom : DIDIER
Prenom : ANDRE
Age : 32
Décision (admis, refusé ou ajourné) : admis
| Candidat numéro 3 : |
Numéro de carte d'identité : 731
Nom : CHAUVET
Prenom : AUGUSTE
Age : 25
Décision (admis, refusé ou ajourné) : refusé
```

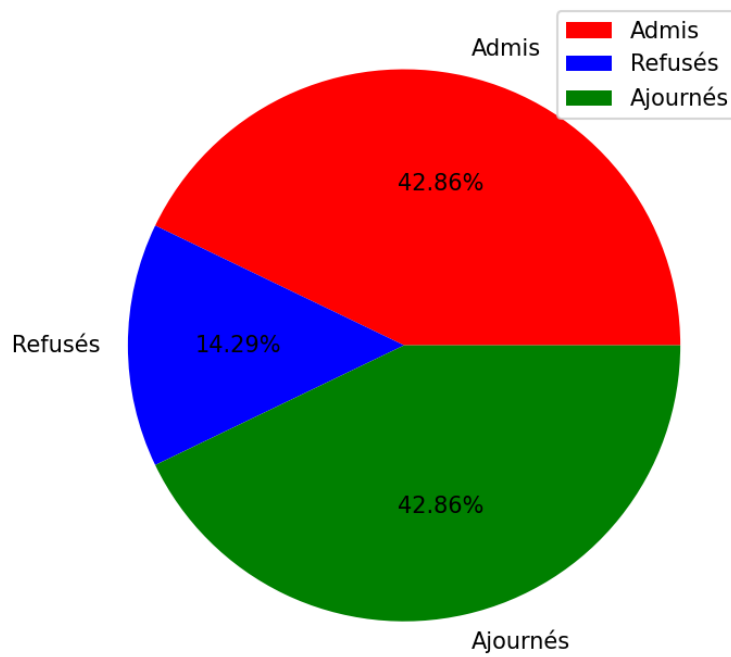
...

```
| Candidat numéro 7 : |  
Numéro de carte d'identité : 198  
Nom : COURTOIS  
Prenom : DAVID  
Age : 20  
Décision (admis, refusé ou ajourné) : ajourné  
| Affichage |
```

```
273;LOPES;ISABELLE;23;admis  
  
734;DIDIER;ANDRE;32;admis  
  
731;CHAUVET;AUGUSTE;25;refusé  
  
738;LEFEBVRE;EMMA;21;ajourné  
  
739;LEMOINE;MAUD;26;ajourné  
  
259;BRUNEL;STEPHANE;26;admis  
  
198;COURTOIS;DAVID;20;ajourné
```

```
| Test de la fonction admis |  
| Affichage |  
273;LOPES;ISABELLE;23;admis  
  
734;DIDIER;ANDRE;32;admis  
  
259;BRUNEL;STEPHANE;26;admis  
  
| Test de la fonction attente |  
| Affichage |  
734;DIDIER;ANDRE
```

```
| Test de la fonction statistique pour la décision ajourné |  
42.857142857142854 %
```



```
| Test de la fonction supprimer |  
Affichage du fichier admis.txt après suppression :  
273;LOPES;ISABELLE;23;admis  
  
259;BRUNEL;STEPHANE;26;admis
```

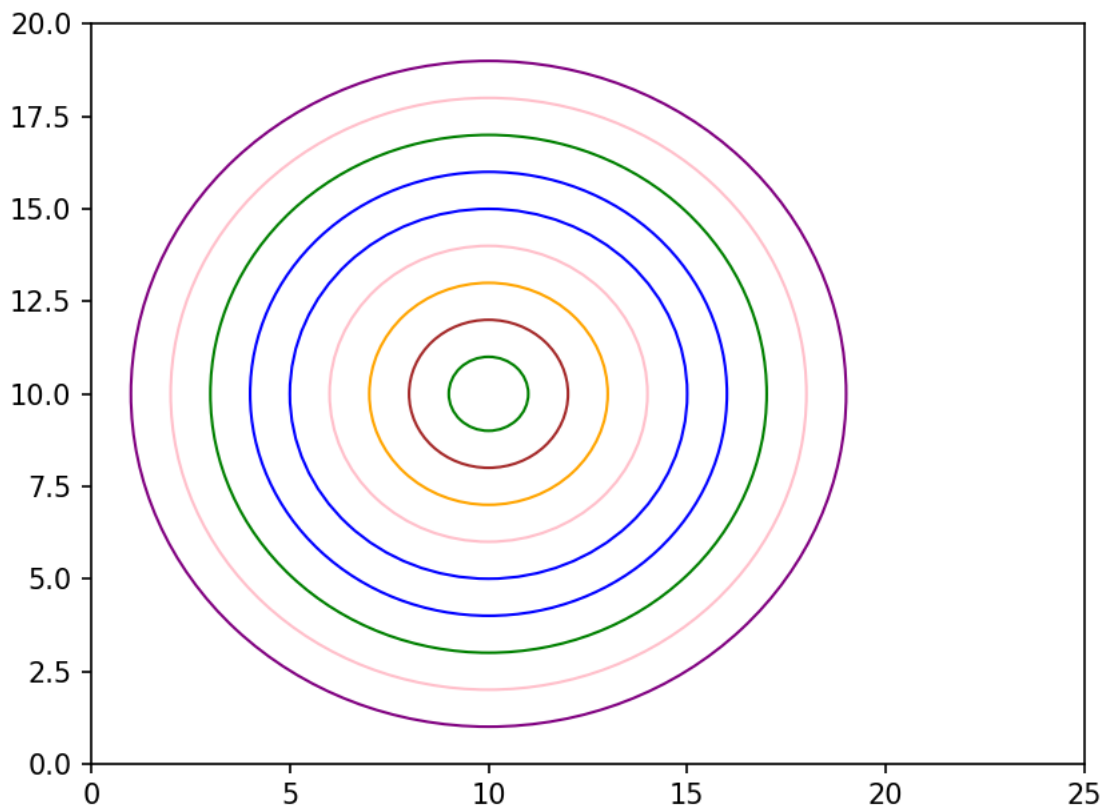
## Exercice 2, Question 1 :

```
def cercle(x, y, r):
    colors = ["blue", "orange", "green", "red", "purple", "brown", "pink", "cyan", "magenta", "yellow"]
    draw_circle = plt.Circle((x, y), r, fill=False, color=colors[randint(0, 8)]) #choix aléatoire de la
    couleur parmi la liste colors
    return draw_circle

from random import *

def main_exo2():
    r = 1
    x = 10
    y = 10
    figure, axes = plt.subplots() #définition des axes
    plt.xlim(0, 25)
    plt.ylim(0, 20)
    for i in range(9):
        draw = cercle(x, y, r)
        axes.add_artist(draw) #tracé du cercle
        r += 1
    plt.show()#affichage du dessin fini
```

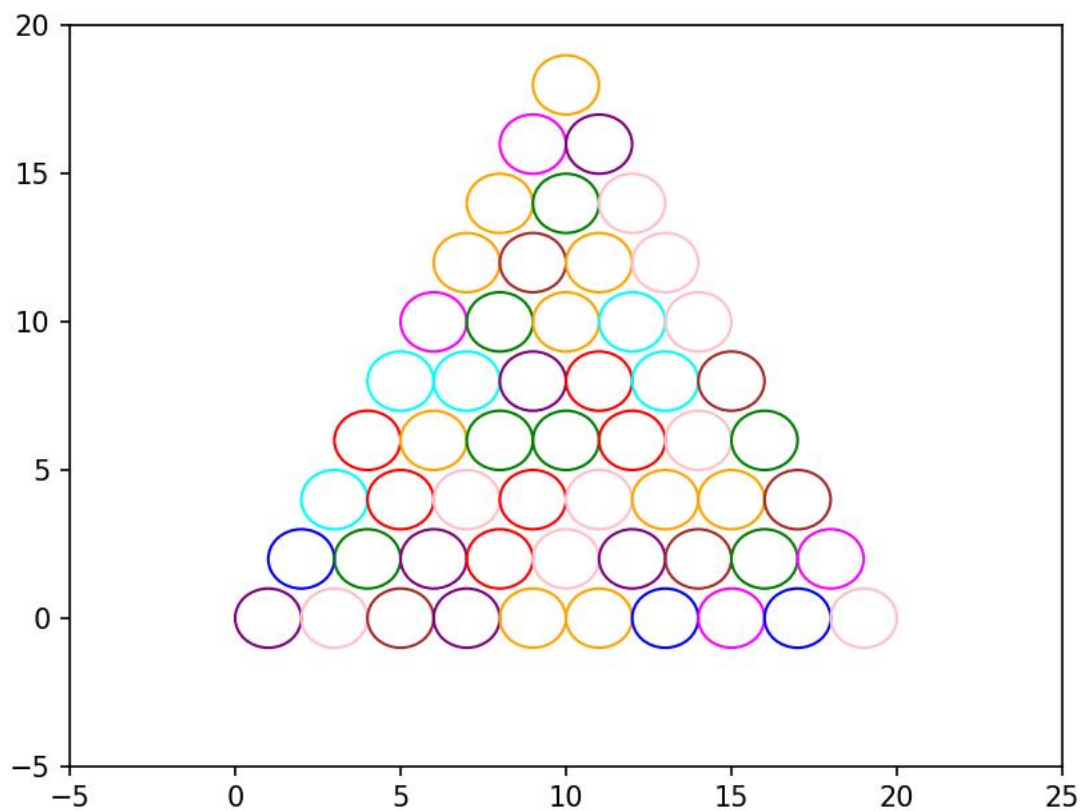
Essai :



## Exercice 2, Question 2 :

```
def pyramide():  
    figure, axes = plt.subplots() #definition des axes  
    r = 1  
    N = 10 #nombre de ligne de la pyramide  
    y = 0  
    x = -r  
    plt.xlim(-5, 25)  
    plt.ylim(-5, 20)  
    for i in range(0, N):  
        for j in range(0, N-i):  
            x = x + 2 * r  
            draw = cercle(x,y, r)  
            axes.add_artist(draw)#tracé du cercle  
            x=i*r  
            y=y+2*r  
    plt.show()#affichage du dessin final
```

Essai :





### Exercice 3 :

```
from math import cos
import matplotlib.pyplot as plt

def ecrire(x_min, x_max):
    #fonction qui permet d'écrire dans le fichier sur chaque ligne x et
    cos(x)
    #on initialise les listes
    x=[]
    cos_x=[]
    #on insère dans les listes x et cos(x) entre x_min et x_max avec un pas
    de 0.1
    while (x_min<=x_max):
        x.append(x_min)
        cos_x.append(cos(x_min))
        x_min+=0.1
    #on crée une liste commune pour les deux, créant des couples (x,cos(x))
    list_c = zip(x, cos_x)
    #on écrit dans le fichier
    with open("math.csv", "w") as fichier:
        for couple in list_c:
            valeur, cos_valeur=couple
            fichier.write(f"{valeur}:{cos_valeur}:\n")

def lire():
    #fonction qui permet de lire le fichier et de faire l'affichage
    graphique
    x=[]
    y=[]
    #on ouvre le fichier en lecture
    with open("math.csv","r") as fichier:
        #on crée une liste contenant toutes les lignes
        lignes=fichier.readlines()
        #pour chaque ligne de la liste
        for ligne in lignes:
            #on crée une liste de chaines de caractère qui étaient séparés
            par ':'
            list=ligne.split(":")
            #on insère la valeur numérique de ces chaines dans les listes
            x.append(float(list[0]))
            y.append(float(list[1]))

    #on crée la courbe
    plt.plot(x,y)
    plt.title("Fonction cosinus")
    #on l'affiche
    plt.show()

def main_exo3():
    ecrire(-5,5)
    lire()

if __name__ == '__main__':
    main_exo3()
```

Essai :

Fonction cosinus

