

FICHE ELEVE

CA SE DEGRADE

1. Un premier algorithme

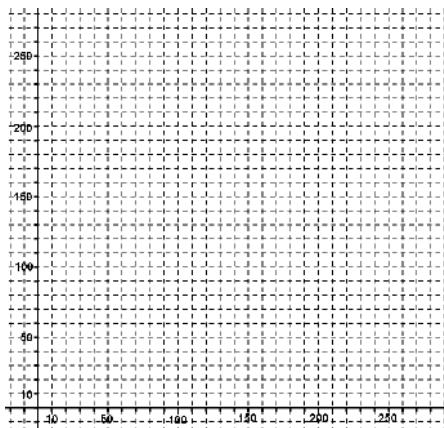
a. Exécuter l'algorithme suivant dans le repère ci-dessous.

Algorithme 1 :

Affecter à a la valeur 0 ;

Tant que $a \leq 200$ faire :

- Tracer le segment dont les extrémités ont pour coordonnées respectives $(50+a, 50)$ et $(50+a, 200)$
- Augmenter a de 10.



b. Avec Python :

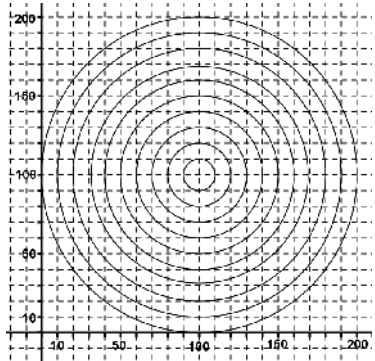
- i. Ouvrir le fichier support_graphique.py
L'instruction `segment` a été créée :
`segment(x1,y1,x2,y2)` permet de tracer un segment d'extrémités les points de coordonnées respectives $(x1,y1)$ et $(x2,y2)$.
- ii. Enregistrer ce fichier sous un nouveau nom dans le répertoire personnel.
- iii. Le script suivant représente la traduction de l'algorithme 1 dans un langage que Python peut interpréter. Ecrire les lignes suivantes dans l'espace prévu à cet effet.

```
a=0
while a<=200:
    segment(50+a,50,50+a,200)
    a=a+10
```

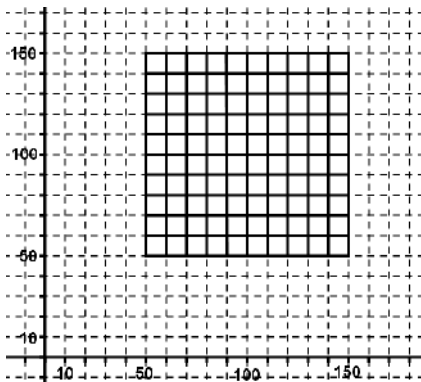
- iv. Exécuter le programme (touche F5)

2. Construction de cercle

- Écrire un algorithme permettant le tracé des cercles suivants :
- L'instruction `cercle(x,y,R)` permet de tracer un cercle de centre le point de coordonnées (x,y) et de rayon R . Tracer ces cercles avec Python. (Effacer les lignes précédemment saisies et écrire votre nouveau programme dans l'emplacement prévu à cet effet).



3. Construction d'une grille



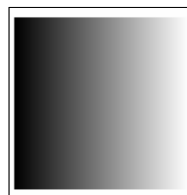
Écrire un programme dont le résultat de l'exécution est la grille ci-contre.

4. Dégradé

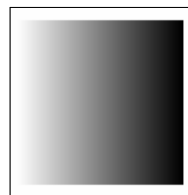
`segment_gris(x1,y1,x2,y2,niveau_de_gris)` permet de tracer un segment dont la couleur est un niveau de gris, nombre compris entre 0 et 100. Plus le niveau de gris est grand plus la couleur est claire. `niveau_de_gris=0` correspond à la couleur noire, `niveau_de_gris=100` correspond au blanc.

- Parmi les propositions ci-après, quelle est celle qui peut représenter le résultat de l'exécution de ce programme :

```
a=0
while a<=100:
    segment_gris(50+a,50,50+a,150,a)
    a=a+1
```

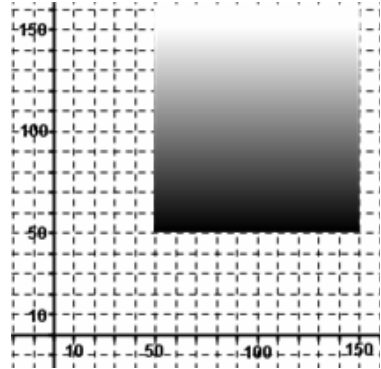


Réponse A



Réponse B

b. Écrire un programme permettant de créer le dégradé suivant :

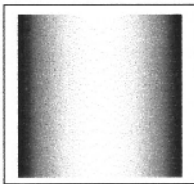


c. On a repris le programme de la question 4a) et on a remplacé le niveau de gris a par $f(a)$ où f désigne une fonction définie sur $[0,100]$ et à valeurs dans $[0,100]$:

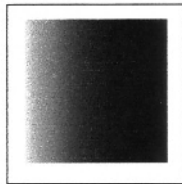
```

a=0
while a<=100:
    segment_gris(50+a,50,50+a,150,f(a))
    a=a+1
    
```

d. Associer à chaque fonction f définie par sa courbe, le dégradé produit par cet algorithme :



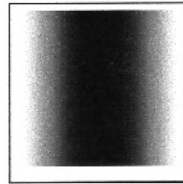
A



B

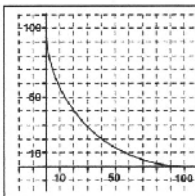


C

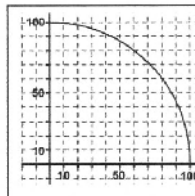


D

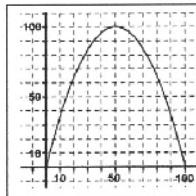
1



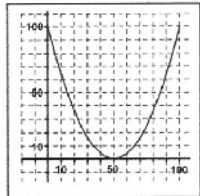
2



3



4



ANNEXE : SUPPORT GRAPHIQUE (langage PYTHON)

```

# Support pour la réalisation d'objets graphiques
from math import *
from tkinter import *
haut_fen=500 # Hauteur de la fenêtre
larg_fen=500 # Largeur de la fenêtre
fenetre=Tk()
can=Canvas(fenetre,bg="white",height=haut_fen,width=larg_fen)
can.pack()

def segment(x1,y1,x2,y2):
    can.create_line(x1+20,haut_fen-(y1+20),x2+20,haut_fen-(y2+20))

def segment_gris(x1,y1,x2,y2,niv):
    can.create_line(x1+20,haut_fen-(y1+20),x2+20,haut_fen-(y2+20),fill="gray"+str(floor(niv)))
def cercle(x1,y1,R):
    can.create_oval(x1+20-R,haut_fen-(y1+20)-R,x1+20+R,haut_fen-(y1+20)+R)

def gris(a):
    return "gray"+str(a)
#Création des axes
can.create_line(20,haut_fen-20,larg_fen,haut_fen-20,arrow=LAST)
can.create_line(20,haut_fen-20,20,0,arrow=LAST)
i=0
while i<larg_fen:
    segment(i,0,i,5)
    can.create_text(i+20,haut_fen-10,text=str(i),anchor="center")
    i=i+50
i=0
while i<haut_fen:
    segment(0,i,5,i)
    can.create_text(10,haut_fen-(i+20),text=str(i),anchor="center")
    i=i+50

#Espace réservé aux lignes de programmation de l'élève

#Fin de l'espace réservé aux lignes de programmation de l'élève

fenetre.mainloop()

```