

## Devoir Seconde

### Première Partie

- 1) Ecrire l'équation réduite de la droite passant par  $A(2 ; -3)$  et  $B(4 ; 2)$ .
- 2) On considère deux points  $A(x_A ; y_A)$  et  $B(x_B ; y_B)$  avec  $x_A \neq x_B$ .  
Justifier que la droite  $(AB)$  a une équation réduite de la forme  $y = mx + p$ .  
Exprimer  $m$  en fonction des coordonnées de  $A$  et de  $B$ .  
Exprimer  $p$  en fonction de  $x_A$ ,  $y_A$  et de  $m$ .
- 3) Compléter l'algorithme ci-dessous où  $x_A, y_A, x_B, y_B, m, p$  sont des nombres :

```
saisir  $x_A, y_A, x_B, y_B$ 
 $m$  prend la valeur .....
 $p$  prend la valeur .....
Afficher " $y=$ ",.....," $x +$  ", .....
```

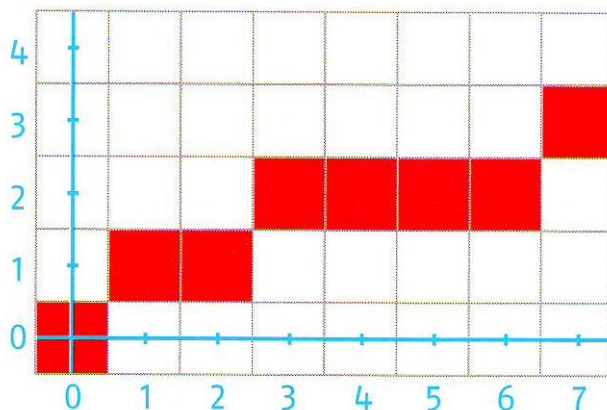
Programmer sur votre calculatrice ou Algobox  
Le tester sur le cas étudié à la question 1).

- 4) Pour aller plus loin :
  - a) Que se passe-t-il sur la calculatrice ou le logiciel si on entre les coordonnées des points  $A(2 ; 1)$  et  $B(2 ; -3)$  ?  
Expliquer pourquoi.
  - b) Proposer un test permettant d'améliorer l'algorithme pour qu'il prenne en compte tous les cas et fournisse toujours l'équation réduite de la droite  $(AB)$ .
  - c) Tester ce nouveau programme et le recopier sur votre copie.

### Deuxième Partie

Sur un écran (de type moniteur d'ordinateur ou de téléphone portable), un point est représenté par un pixel, c'est-à-dire un petit carré. Ce pixel peut être allumé ou éteint.

Pour représenter un segment, on allume une suite continue de pixels. Deux pixels doivent être reliés par un côté ou par un sommet, comme le montre le schéma ci-contre.



Chaque pixel est repéré par son numéro de colonne (abscisse) et son numéro de ligne (ordonnée).

On souhaite trouver un algorithme donnant les pixels à allumer pour construire un segment.

Les extrémités du segment seront le pixel  $A$  repéré ci-dessus par l'origine du repère et le pixel  $B(x_B; y_B)$  avec  $y_B \leq x_B$ .

- 1) On choisit  $B(10;7)$ 
  - a) Tracer le repère sur une feuille quadrillée et placer les points  $A$  et  $B$ .
  - b) En prenant un seul pixel par colonne, relier les points  $A$  et  $B$ .  
Proposer au moins deux autres solutions sur d'autres quadrillages.
  - c) Quelle solution vous semble la meilleure ?
- 2) On considère la fonction  $f$  linéaire représentée par la droite  $(AB)$ .
  - a) Déterminer la fonction  $f$ . Tracer le segment  $[AB]$ .
  - b) Le choix fait en 1) c) semble-t-il toujours pertinent ? Justifier la réponse.

### 3) L'algorithme de Bresenham

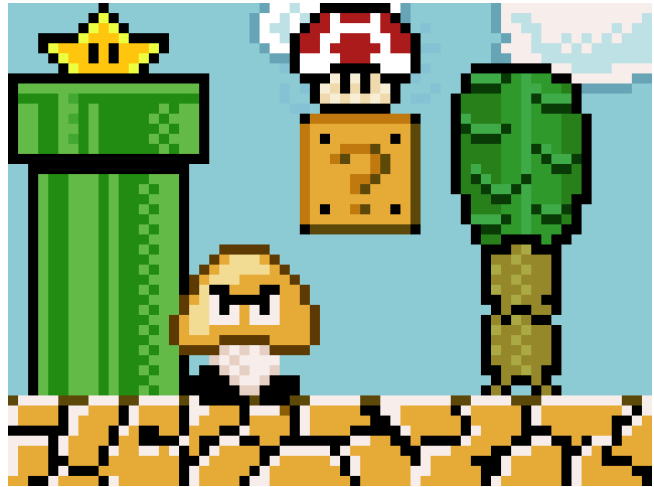
Pour chaque valeur entière de  $x$ , on calcule  $f(x)$  en arrondissant à l'entier le plus proche pour obtenir les coordonnées du pixel à allumer.

a) A l'aide de la formulation de l'algorithme de Bresenham, déterminer les pixels à allumer.

b) Proposer un algorithme demandant les coordonnées des deux points  $A$  et  $B$  et utilisant l'algorithme de Bresenham pour calculer les coordonnées des pixels à allumer.

c) Peut-on, par cette méthode, représenter tous les segments ?

(Vous relirez précisément l'énoncé et donnerez un ou plusieurs contre-exemples).



### 4) Mise en œuvre de l'algorithme avec Python

Dans un premier temps, demander à l'utilisateur les coordonnées **entières** des points  $A$  et  $B$  dont les abscisses seront différentes.

Déterminer le coefficient directeur et l'ordonnée à l'origine de la droite ( $AB$ ) que vous affecterez à des variables de votre choix.

Afin de parcourir toutes les abscisses entre les deux points, permuter les coordonnées de  $A$  et de  $B$  si l'abscisse de  $A$  n'est pas strictement inférieure à celle de  $B$ .

Pour permettre le tracer de courbes et de formes il sera nécessaire d'utiliser les instructions plot et show de la bibliothèque matplotlib.pyplot :

```
from matplotlib.pyplot import plot,show
plot([xA,xB],[yA,yB]) permet de tracer le segment [AB].
```

L'instruction `round(x,p)` de Python permet d'arrondir le flottant  $x$  avec  $p$  chiffres après la virgule. En particulier, `round(x,0)` arrondit le flottant  $x$  à l'entier le plus proche.

Compléter la première partie de votre programme par le code ci-dessous permettant de parcourir toutes les abscisses entre celle de  $A$  et celle de  $B$  pour calculer l'ordonnée du pixel selon l'algorithme de Brésenham et enfin afficher le pixel correspondant.

```
for i in range(... , ...):
    y=round(... , ...)
    plot(i,y,marker='s',markerfacecolor='red', markersize=50)
show()
```

Utiliser ce programme pour retrouver votre construction du 3). Vous aurez peut-être à modifier la taille de la fenêtre ou bien la taille du carré par le champ `markersize` afin d'avoir un rendu plus agréable à l'œil.

Joindre une impression du programme et la figure obtenue à votre devoir ou bien envoyer ces deux éléments à [frederic.vivien@ac-rouen.fr](mailto:frederic.vivien@ac-rouen.fr) en complément de votre devoir papier.

```
xA=int(input('Abscisse de A : '))
yA=int(input('Ordonnée de A : '))
xB=int(input('Abscisse de B : '))
yB=int(input('Ordonnée de B : '))
```

```
m=(yB-yA)/(xB-xA)
p=yA-m*xA
```

```
from matplotlib.pyplot import plot,show
```

```
if xA>xB:
    xA,xB=xB,xA
    yA,yB=yB,yA
```

```
plot([xA,xB],[yA,yB])
```

```
for i in range(xA,xB+1):
    y=round(m*i+p,0)
    plot(i,y,marker='s',markerfacecolor='red', markersize=50)
```

```
show()
```

