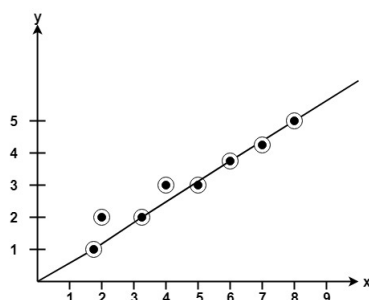


TP n°2 : Droites discrètes, Bresenham et extensions

Dans ce TP, on va chercher à tracer des droites discrètes allant du point $(0,0)$ à un certain point (dx, dy) . On parlera alors de droite de *pente* (dx, dy) .



1 Algorithme de Bresenham

Vous trouverez dans le fichier *droites.c* sur Moodle plusieurs fonctions :

- *affiche* qui permet d'afficher un plan, vu comme un tableau 2D de caractères de taille MAX.
- *droite_triviale* qui permet de calculer les coordonnées des pixels pour tracer la droite discrète de pente (dx, dy) .
- *droite_triviale_verif* qui permet de réaliser le même calcul que précédemment mais également d'afficher les points obtenus dans le terminal, en ajoutant un caractère `.` dans `p1` lorsque nécessaire.

Dans la fonction *main*, vous trouverez des tests de ces fonctions pour les droites de pentes $(11, 3)$ et $(24, 10)$. Votre tâche est :

1. d'implémenter l'algorithme de Bresenham vu en cours. Il vous faut écrire des fonctions *droite_br* et *droite_br_verif*; la première calculant les coordonnées des points de la droite et la seconde affichant en plus les points correspondant.
2. de tester cette fonction sur les droites de pente $(11, 3)$ et $(24, 10)$. Obtenez-vous le même résultat que pour l'algorithme trivial ? Si non, pouvez-vous l'expliquer et le corriger ?
3. de tester, pour toutes les droites de pente (dx, dy) avec $0 \leq dx \leq 100$ et $0 \leq dy \leq 100$, le temps de calcul des droites discrètes en utilisant l'algorithme trivial et l'algorithme de Bresenham.

2 Algorithme du pas de deux

Dans l'archive *Articles.zip* sur Moodle, vous trouverez les articles de Rokne, Wyvill & Wu (pas de deux) et de Graham & Iyengar (pas de trois). En vous inspirant de ces articles, écrire des fonctions *droite_rww* et *droite_rww_verif* qui permettent de calculer la droite discrète de pente (dx, dy) en utilisant l'algorithme du pas de deux. Comparer les résultats avec les fonctions précédentes, et ajouter la mesure du temps d'exécution de l'exemple précédent avec cette nouvelle fonction.

3 Algorithme d'Angel & Morrison

1. Écrire une fonction permettant de calculer le PGCD des entiers dx et dy en utilisant l'algorithme d'Euclide.
2. Écrire une fonction, prenant en paramètre les entiers dx et dy , et un entier k , et qui permet de tracer k fois la droite de pente (dx, dy) .
3. Implémenter l'algorithme de Angel & Morrison : on calculera le PGCD d de dx et dy , puis on tracera la droite de pente $(dx/d, dy/d)$ à l'aide de l'algorithme de Bresenham, que l'on répètera d fois.

4 Un algorithme sur les mots de trace

Dans cette section, vous devrez implémenter l'algorithme de votre choix parmi :

1. Dulucq & Bourdin;
2. Green & Pitteway;
3. Berstel.

Il faudra écrire les fonctions nécessaires aux opérations requises sur les mots de trace d'une droite, puis tracer la droite en fonction du mot de trace final obtenu.