

## TD n° 5 Géométrie Algorithmique

### Préparation de code source en C++ (mais vous pouvez aussi le faire en java ou python)

Vous implémenterez les classes suivantes :

- Point pour implémenter un Point 2D qui hormis les méthodes de base (constructeurs/destructeur, getter/setter, affichage) disposera des méthodes suivantes : void genererPointAleatoire(int borneX, int borne Y), double distanceEuclidienne(Point p), surcharge des opérateurs > et < (en cas d'égalité entre les ordonnées on compare les abscisses et inversement).
- Vecteur, caractérisé par 2 points 2D p0 et p1, qui hormis les méthodes de base (constructeurs/destructeur, getter/setter, affichage) disposera des méthodes suivantes : double calculPente(), puissance (Point m) qui calcule la puissance de m , int produitScalaire(Vecteur v), int determinant (Vecteur v), une surcharge de l'opérateur < qui permet s'appuyer sur la valeur de la pente .
- NuagePoints, caractérisé par un vector de Point , qui hormis les méthodes de base (constructeurs/destructeur, getter/setter, affichage) disposera des méthodes suivantes : genereNuagePointAlea( int nbPoints, int borneX, int borneY), d'une méthode permettant de copier un nuage de points, une méthode permettant d'ajouter/retirer, un point au nuage de points, une méthode permettant de trier le nuage de Points, une méthode permettant d'afficher les coordonnées des points du nuage, une méthode permettant de faire un affichage graphique des points du nuage.
- Polygone, caractérisé par un contour de type NuagePoints, qui hormis les méthodes de base (constructeurs/destructeur, getter/setter, affichage) disposera des méthodes suivantes : triPolaire, printGraphique. Cette classe sera enrichie au fur et à mesure des Tds  
Vous pouvez par exemple rajouter une méthode qui permet d'implémenter l'algorithme de fenêtrage étudié dans le TD n°3.

**NB :** Pour éviter de développer une interface graphique et si vous développez en C++, vous tracerez les points ou segments dans un fichier postscript.

Voici un exemple de programme qui aboutit à la génération du fichier postscript, qui pourra être converti en pdf et qui pourra être visualisé par n'importe quel viewer de postscript par exemple `ggv`, `kghostview...`

```
const int offset=1;

ofstream output ("Points.ps");//
output << "%!PS-Adobe-3.0" << endl;
output << endl;

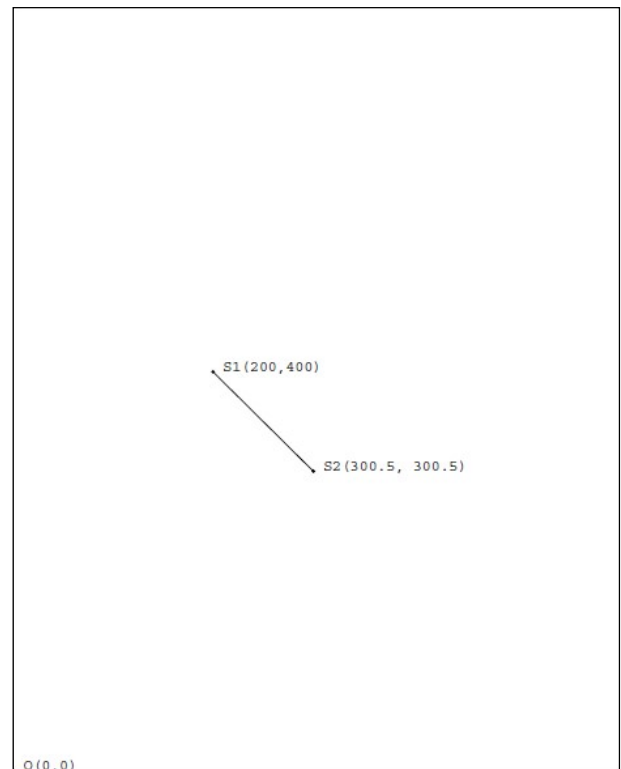
//on trace le point 0(0,0)
output << 0 << " " << 0 << " 2 0 360 arc" <<endl; //on trace un cercle en coordonnées (0,0)
output << "0 setgray" <<endl; //on choisit le noir
output << "fill" <<endl; //on remplit le disque
output << 0+offset << " " << 0+offset << " moveto" <<endl;
//on déplace le curseur en (0+offset, 0+offset)
output << "/Courier findfont 15 scalefont setfont" << endl; // on choisit les paramètres de la fonte
output << "( 0(0,0) )" << "show" << endl; //on écrit 0(0,0)
output << "stroke" << endl; //on exécute les commandes précédentes pour imprimer la figure conçue
output << endl;

//on trace le sommet S1(200,400)
output << 200 << " " << 400 << " 2 0 360 arc" <<endl;
output << "0 setgray" <<endl;
output << "fill" <<endl;
output << 200+offset << " " << 400 << " moveto"
<<endl;
output << "/Courier findfont 15 scalefont setfont" <<
endl;
output << "( S1(200,400) )" << " show" << endl;
output << "stroke" << endl;
output << endl;

//on trace le sommet S2(300.5,300.5)
output << 300.5 << " " << 300.5 << " 2 0 360 arc"
<<endl;
output << "0 setgray" <<endl;
output << "fill" <<endl;
output << 300.5+offset << " " << 300.5 << " moveto"
<<endl;
output << "/Courier findfont 15 scalefont setfont" <<
endl;
output << "( S2(300.5, 300.5))" << " show" << endl;
output << "stroke" << endl;
output << endl;

//on trace un segment entre s1 et s2
output << 200 << " " <<400 << " moveto" << endl;
output << 300.5 << " " << 300.5 << " lineto" << endl;
output << "stroke" << endl;
output << endl;

output.close();
```



**Figure 1:** Visualisation du fichier pdf obtenu à partir de la conversion du fichier postscript Point.ps (généré à partir du code source ci-contre).

## Exercice : Enveloppe Convexe

- Vous implémenterez l'algorithme de Graham en C++ ou Python à l'aide des classes préparées dans la partie précédente
- Vous implémenterez toujours à l'aide des mêmes classes la version par division et fusion