# LO27 - Report GeometricLib

We don't think that our function are the most optimize version and we suggest ourselves some improvements sometimes.
To preserve a readable document we add link to algorithm in .txt format.

## 1. CreatePoint

### 1.1. Algorithm :

Click here to open algorithm

### 1.2. Explanations :

This isn't a complex algorithm of course. It consist in create a point with the coordinates which are given by user. The function return a new nice point !

## 2. CreatePolygon

### 2.1. Algorithm :

Click here to open algorithm

### 2.2. Explanations :

This function create a new empty polygon and need no arguments.

## 3. addPoint

### 3.1. Algorithm :

Click here to open algorithm

### 3.2. Explanations :

This function add a given point in a given polygon at a given index. So we use a for loop to find the point just before the right index and we insert the new one. Thus we can manage all link.

## 4. addTail

### 4.1. Algorithm :

Click here to open algorithm

### 4.2. Explanations :

This function seems to addPoint but add a point at the end of a polygon.

## 5. removePoint

### 5.1. Algorithm :

Click here to open algorithm

### 5.2. Explanations :

This function is the opposite of addPoint : it removes a given point in a given polygon at a given index.

## 6. unionPolygons

### 6.1. Algorithm :

Click here to open algorithm

### 6.2. Explanations :

This function returns a Polygon which is the addition of two incoming Polygon. Because of the complexity of some case we decided to make a convexhull of at the end to have a functional version.

## 7. SegmentsCross

### 7.1. Algorithm :

Click here to open algorithm

### 7.2. Explanations :

This function return a intersection type which contain the coordinates of the four point which represent the two crossing segment and the coordinates of the intersection point. It's a function we re-use in different others functions.

## 8. IntersectionPolygons

### 8.1. Algorithm :

Click here to open algorithm

### 8.2. Explanations :

This function returns a Polygon which is the representation of a common surface.
We present here just bêta-test algorithm because the C process doesn't work. Small research show us that is possible but time is over now :)
Links :

- http://geomalgorithms.com/a09-_intersect-3.html
- http://en.wikipedia.org/wiki/Line_segment_intersection

## 9. exclusiveORPolygons

### 9.1. Algorithm :

No readable algorithm because of lack of time.

### 9.2. Explanations :

This function can be build with union and differencePolygon and returns a Polygon which is the representation of

## 10. differencePolygons

### 10.1. Algorithm :

No readable algorithm because of lack of time.

### 10.2. Explanations :

This function returns the surface which is the remaining polygon if we subtract the second incoming polygon to the first.

## 11. containsPoint

### 11.1. Algorithm :

Click here to open algorithm

### 11.2. Explanations :

This function returns a boolean : true if a point is contained into a polygon, false otherwise.
We exclude the case when the virtual segment go on one of the two points of a segment of the polygon (otherwise the point would be counted 2 time)

## 12. containsPolygon

### 12.1. Algorithm :

Click here to open algorithm

### 12.2. Explanations :

This function returns a Status enumeration type that could take the following values:

- INSIDE if the second polygon is fully inside the first one
- OUTSIDE if the second polygon is fully outside the first one
- INTERSECT if the second polygon is partially inside/outside the first one, in other words intersecting the second one
- ENCLOSING if the first polygon is fully in
- EQUAL if polygons are strictly the same

We consider that two polygons are equals if they have exactly the same size, and same point which same coordinates.

## 13. centralSymetry

### 13.1. Algorithm :

Click here to open algorithm

### 13.2. Explanations :

This function which return a Polygon, compute the central symmetry of  a specified polygon according to a reference point given by the user.

## 14. rotatePolygon

### 14.1. Algorithm :

Click here to open algorithm

### 14.2. Explanations :

This function return a Polygon which is the rotation a the incoming Polygon according to the angle given by the user. We use complex numbers because the module stay the same and it's easy to change the argument and fall back to coordinates.

## 15. translatePolygon

### 15.1. Algorithm :

Click here to open algorithm

### 15.2. Explanations :

This function translate a Polygon from a first point to a second : this two points define a vector which is the direction of the translation. After two test we decided to adopt the final method which is consisted in add x and y coordinates of the vector to each point in the polygon and we return the translated polygon.

## 16. scalePolygon

### 16.1. Algorithm :

Click here to open algorithm

### 16.2. Explanations

This function extend a polygon in proportion of the given factor. We decided to take the point 0 to reference point so we can multiplicate directly each coordinates by the factor and as a result we returns a Polygon.

## 17. convexhullPolygon

### 17.1. Algorithm :

Click here to open algorithm

### 17.2. Explanations :

convexhullPolygon help in different other function. It compute the convex outline of a given polygon. We consider that the polygon is a mathematical geometric shape that contains three or more points. So the 2 first points are added to the new polygon so we have a first segment and we use the vectorial product to be sure we turn always in the same direction. So we return a new polygon which contain a convex hull of the incoming polygon.

## 18. printPoint

### 18.1. Algorithm :

It's not necessary to build a algorithm for this.

### 18.2. Explanations :

This procedure consisted in print the coordinates of a point on the terminal. If we would use a graphical interface we could have a picture on a orthogonal graph. We decide to display just two decimal to be sure that is readable and it’s useless in this case to have more.

## 19. printPolygon

### 19.1. Algorithm :

Click here to open algorithm

### 19.2. Explanations :

We re-use the printPoint function here to print on the terminal all coordinates of a Polygon. We respect the format [[x1,y1],[x2,y2], ...,[xn,yn]]. Finally printPolygon is a procedure.

## 20. toString

### 20.1. Algorithm :

Click here to open algorithm

### 20.2. Explanations :

This function can return a string which contains all coordinates of a Polygon like printPolygon so we respect the format [[x1,y1],[x2,y2], ...,[xn,yn]]. Knowing the structure we could analyse it to deduct or display something.
WARNING ! : We use a buffer so we can't manage double which are greater than the length of the BUFFER. That can be improved.