

Tutorial 1

1.1 Points and polygon (record)

We want to compute properties of some polygon (perimeter, average edge length, bounding box,...). The bounding box is a rectangle enclosing the polygon with sides parallel to the axes. The vertices of the polygon are given (in the order of the perimeter route) by real coordinates in a plane and the name of the point (char).

The list of coordinate is supposed correct and is only read once to store the polygon at the beginning. An example of list of point is the following :

- 1,5 3,7 A
- 6,4 4,6 B
- 5,8 8,3 C
- ...
- 1,5 3,7 A (last vertex, do not add it to the data structure)

The coordinates and the name of the last vertex are the same as those of the first point. This vertex only allows us to detect the closing of the polygon. It should not be entered into the data structure.

- **Propose a data structure to store the polygon,**
- **Write the main algorithm to read a polygon, compute properties and print them (just by calling secondary functions, without writing the content of these functions),**
- **Write the content of the secondary functions to compute the perimeter, the average edge length, the bounding box.**

1.2 Queen and pawns

We consider an 8 by 8 chessboard containing 8 pawns and a queen. The queen's position is just its two coordinates and the positions of the pawns will be stored in an 8 by 8 matrix. We will define them in the following :

```
int [8][8] chessBoard; // this is a boolean matrix, with only 0s and 1s

int qx, qy; // the queens coordinates
```

The queen can move on the lines (horizontal, vertical and diagonal) on which she is standing. If two pawns or more are over the same line, only the first is in the capture. For a pawn, to be in the capture means that the queen can capture it in her next move.

- **Write the main algorithm to compute this problem. Try to use only one secondary function `nbr_pawns` to compute the number of pawns in each line.**
- **Write the content of this function.**

There are different ways to write these functions, here is one, with two different approaches for the secondary function

1.3 Recursion

Compute using a recursive algorithm :

- the maximum of an integer array
- the reverse printing of an integer array
- a function to reverse a word and another to check if the word is a palindrome (a word which reads the same forward and backward, such as madam, radar, level,...)