CAD practical – programming part 8:
Triangulating a monotone polygon

The zip file "CADCG_W8_Code.zip" is available on the web page. It contains the code needed for generating the executable. The program "polygon_triangulation" which is the topic of the present session. To compile this executable, the Build Target is "`poly_tri-static`".

This code can be built directly but you will only see the monotone polygon without partition.

Monotone polygon triangulation

Here, the aim is to triangulate a monotone polygon with respect to the x-direction (Theory: ppt. 7 pp. 37-51). The polygon is described by an edge-vertex data-structure (see code) with an anti-clockwise orientation.

In the file "test/polygon_triangulation/main.cc", implement the two following functions:

- `bool check_diagonal(const vertex &_v, const vertex &_w)`
  It checks if a line joining the vertices `_v` and `_w` lies inside the polygon. This can be done be checking the oriented angles* between the line _v - $\_v_1$ and the successive lines _v - $\_v_n$ (where $\_v_1$ is the next neighbouring vertex following `_v` in anti-clockwise order, and $\_v_n$ is the n-th next neighbouring vertex in anti-clockwise order), till $\_v_n$ == `_w`. (See the presentation for details).

  *The function `compute_angle` does that for you.

- `void make_triangulation(std::vector<edge> const &polygon_0, std::vector<edge> &polygon)`
  It implements the triangulation algorithm itself for a monotone polygon along the x-direction. The algorithm is described on page 48 of the course CADCG_07. Note that provided code already tags for you the vertices as belonging to the upper or lower chain (see function `tag_vertices`).

In order to check the correctness of your code, you can compare your results with what is shown in slides 37 to 47 of the course ppt 7. Or change the polygon to compare your results with the one of the slides if this session.