Sutanu Bera
Roll - CSE/22107/0961
Algorithm II Lab (CSC711)
Lab - 05

PROGRAM FILES LINK:

Polygon Monotonicity & Triangulation (Ear Clipping Method): CLICK

ALGORITHM:

1. Checking Polygon Monotonicity

A polygon is monotone with respect to an axis (say x or y) if every line perpendicular to that axis intersects the polygon boundary at most two times.This means the polygon can be split into two monotone chains.

Steps to check monotonicity (for x or y axis):

1. **Input**: A polygon represented as a sequence of vertices in order.
2. Choose an axis (x or y).
3. Traverse the polygon edges in order and compute the **direction of movement** along that axis:
    - `+1` → increasing (right/up)
    - `-1` → decreasing (left/down)
    - `0` → flat (ignore).
4. Compress consecutive same directions into one (remove repeats like +1,+1).
5. Count the number of **direction changes**.
6. If the number of changes = **2**, then polygon is **monotone** with respect to that axis.
7. Otherwise, it is **non-monotone**.

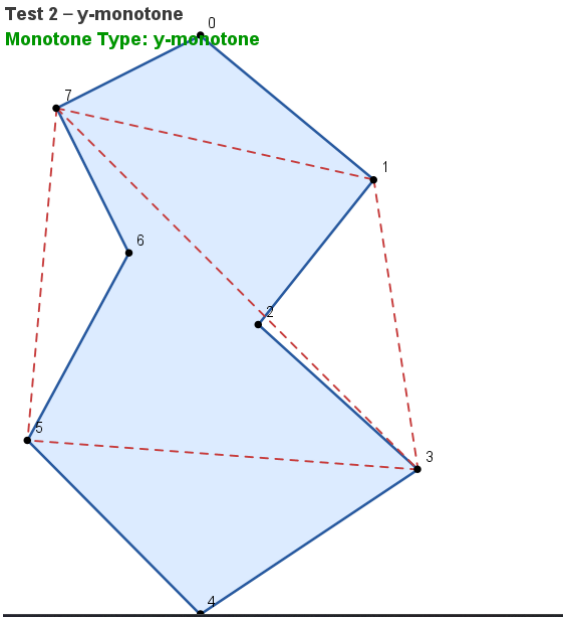
2. Triangulation (Ear Clipping Method)

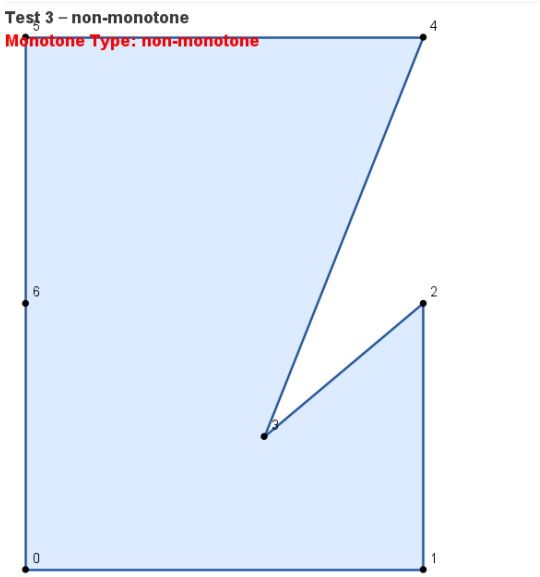
A **triangulation** splits the polygon into non-overlapping triangles using diagonals.

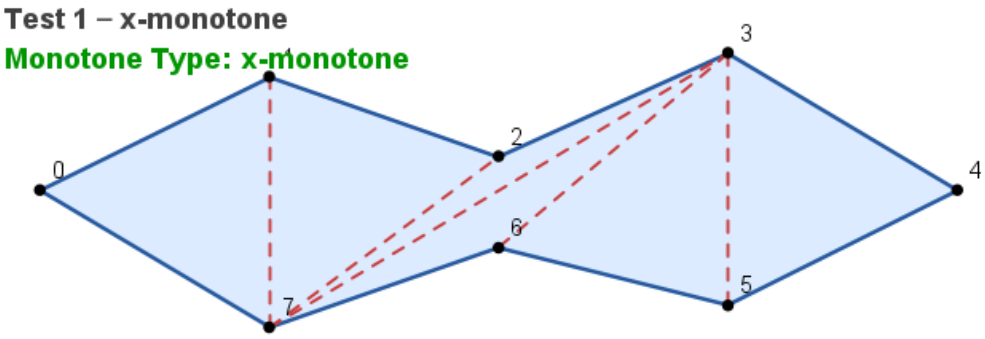
Ear Clipping Steps:

1. **Input**: A simple polygon (must be monotone for correctness).
2. Ensure polygon vertices are ordered **counter-clockwise (CCW)**. If clockwise, reverse.
3. Store vertices in a list.
4. While polygon has more than 3 vertices:
    - Pick a vertex $\mathbf{v_i}$ and its neighbors ($\mathbf{v_{i-1}}$, $\mathbf{v_i}$, $\mathbf{v_{i+1}}$).
    - Check if it forms a **convex angle** (CCW orientation).
    - Ensure no other vertex of the polygon lies **inside** this triangle.
    - If both conditions true → it's an **ear**.
    - Clip the ear:
        - Add diagonal ($\mathbf{v_{i-1}}$, $\mathbf{v_{i+1}}$) to triangulation list.
        - Remove vertex $\mathbf{v_i}$ from polygon.
5. Continue until only one triangle remains.
6. Output the diagonals (triangulation).

INPUT AND OUTPUTS:







```
Test 1 ? x-monotone | x-monotone=true y-monotone=false
Triangulation produced 5 diagonals (5 expected).
Test 2 ? y-monotone | x-monotone=false y-monotone=true
Triangulation produced 5 diagonals (5 expected).
Test 3 ? non-monotone | x-monotone=false y-monotone=false
Polygon is non-monotone; triangulation skipped.
```