POLYGON CLIPPING

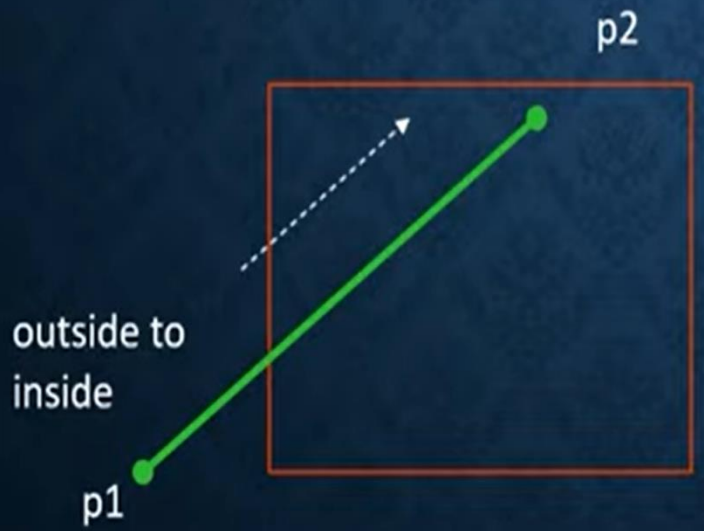
SUTHERLAND AND HODGEMAN POLYGON CLIPPING

SUTHERLAND HODGEMAN ALGORITHM

- Polygon is a set of lines joined together
- Polygon clipping has 4 stages
  - Left clip
  - Right clip
  - Top clip
  - Bottom clip
- For each stage there are **4 cases** to be checked for

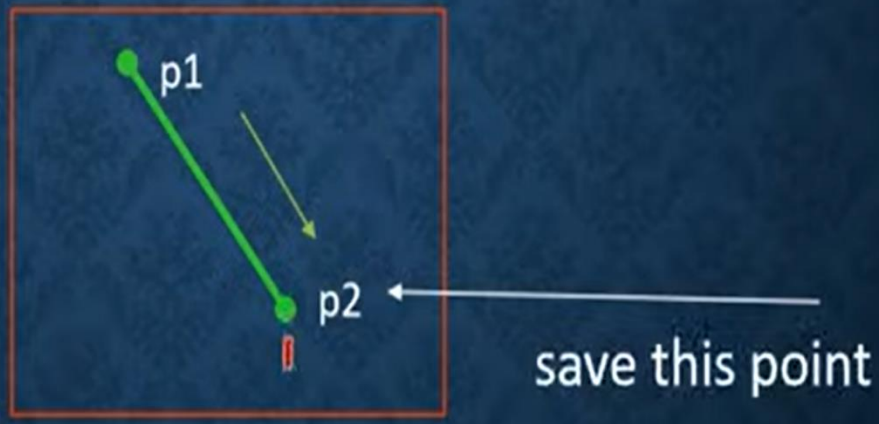
4 CASES?

1. if moving from...
   ...outside to inside
      reject the start point
      & save the intersection point on window boundary
      and the vertex



4 CASES?

2. If moving from...
   ...inside to inside
      save second vertex



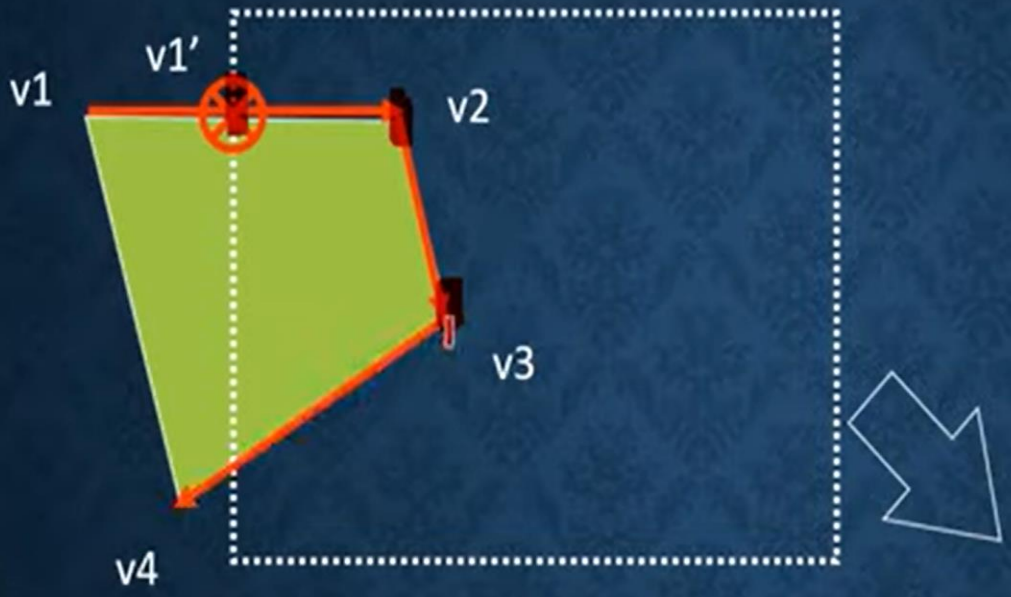
4 CASES?

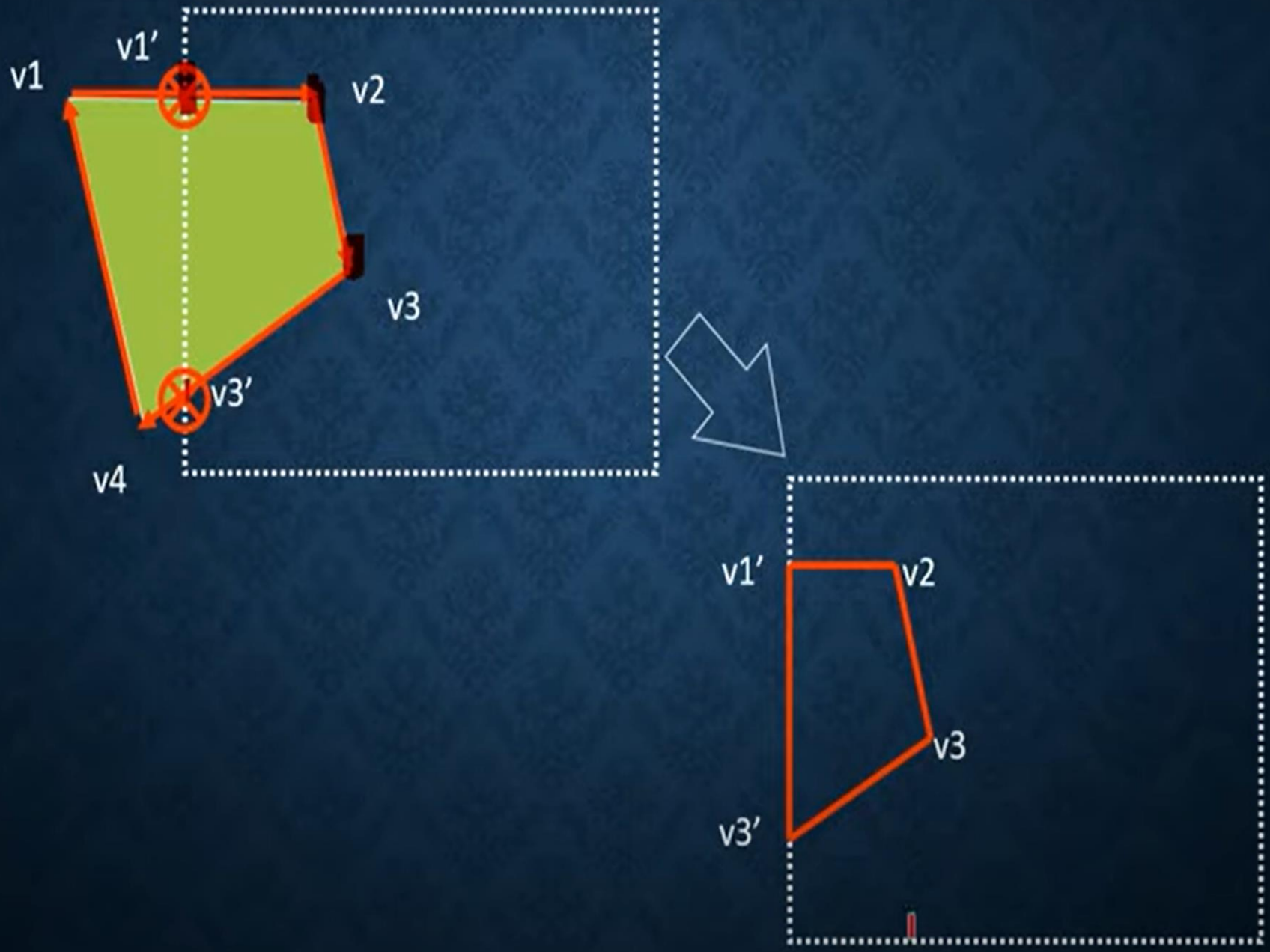
3. if moving from...
   ...inside to outside
      save intersection point
      reject the end point

4 CASES?

4. if moving from...
   ...outside to outside
      save none

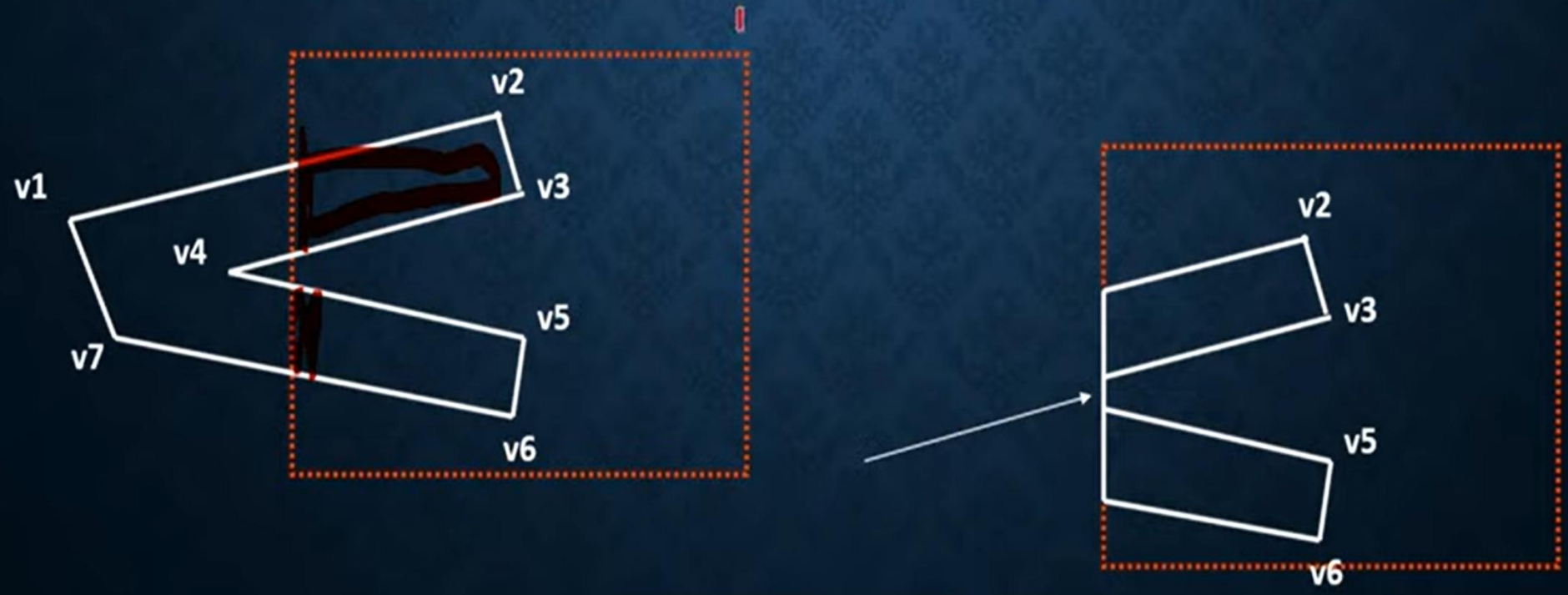
EXAMPLE


v1
v1’
v2
v3
v4

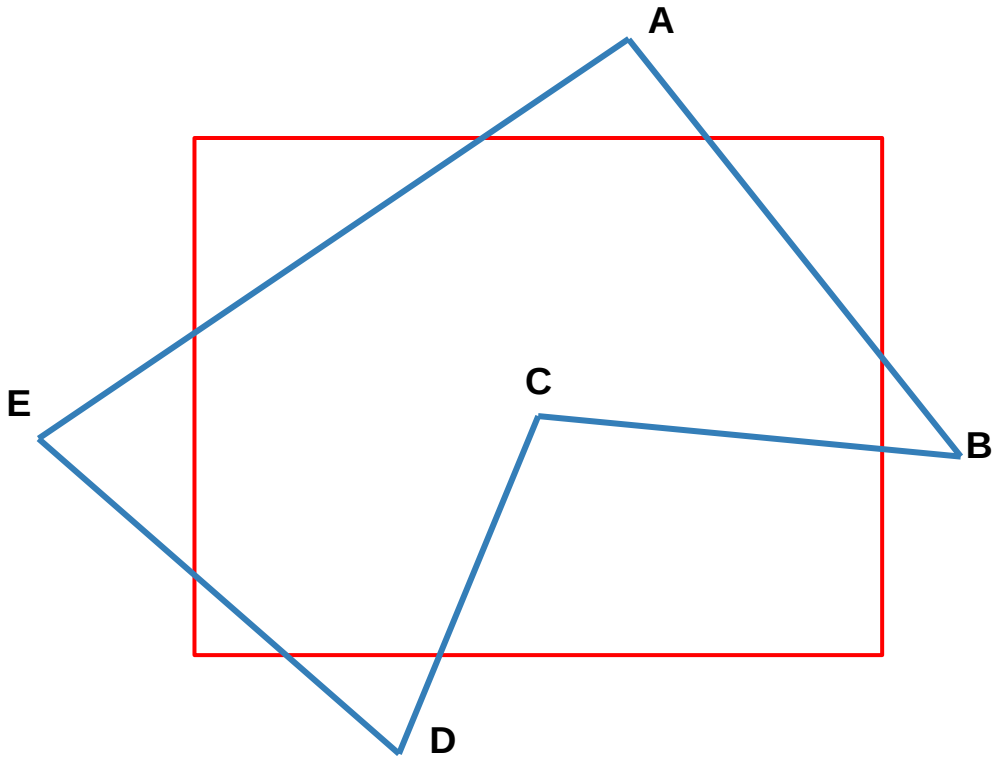

v1
v1’
v2
v3
v3’
v4
v1’
v2
v3
v3’

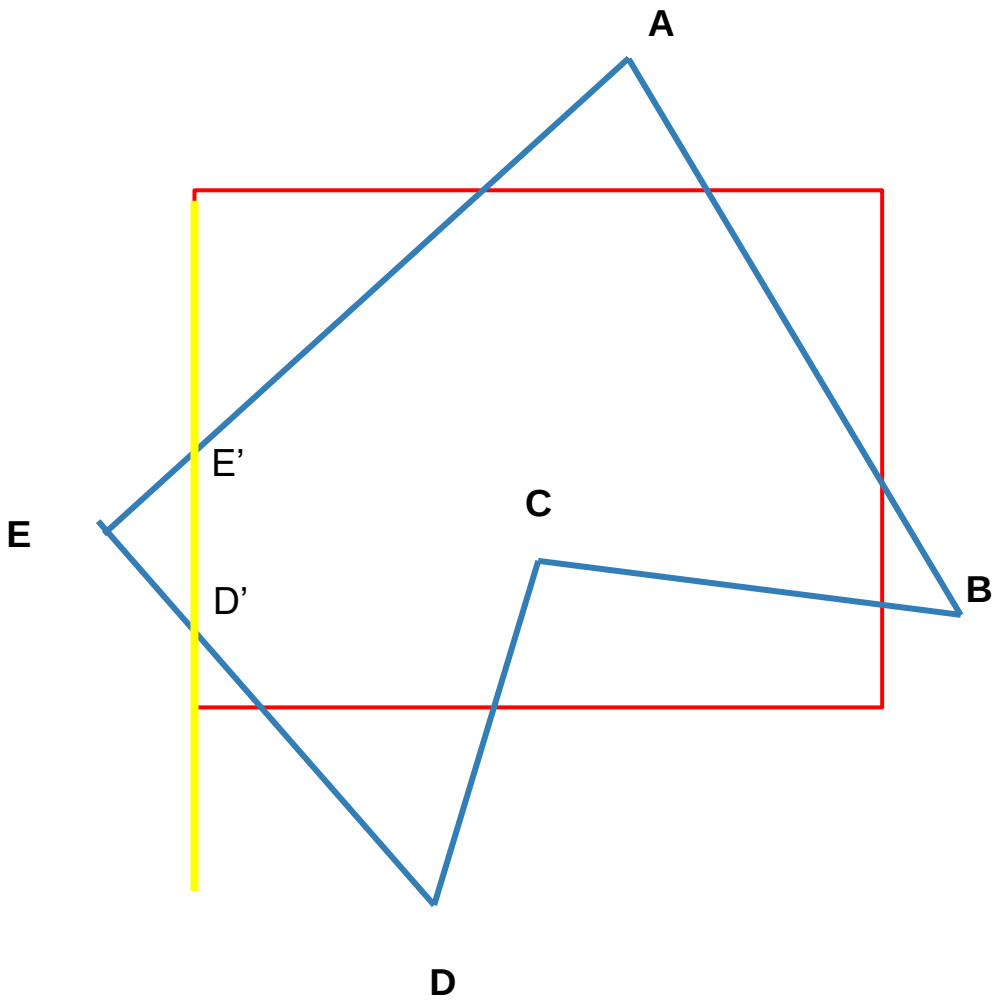
DISADVANTAGE

- This algorithm will not clip the concave polygon properly
- A line is created through the window boundary

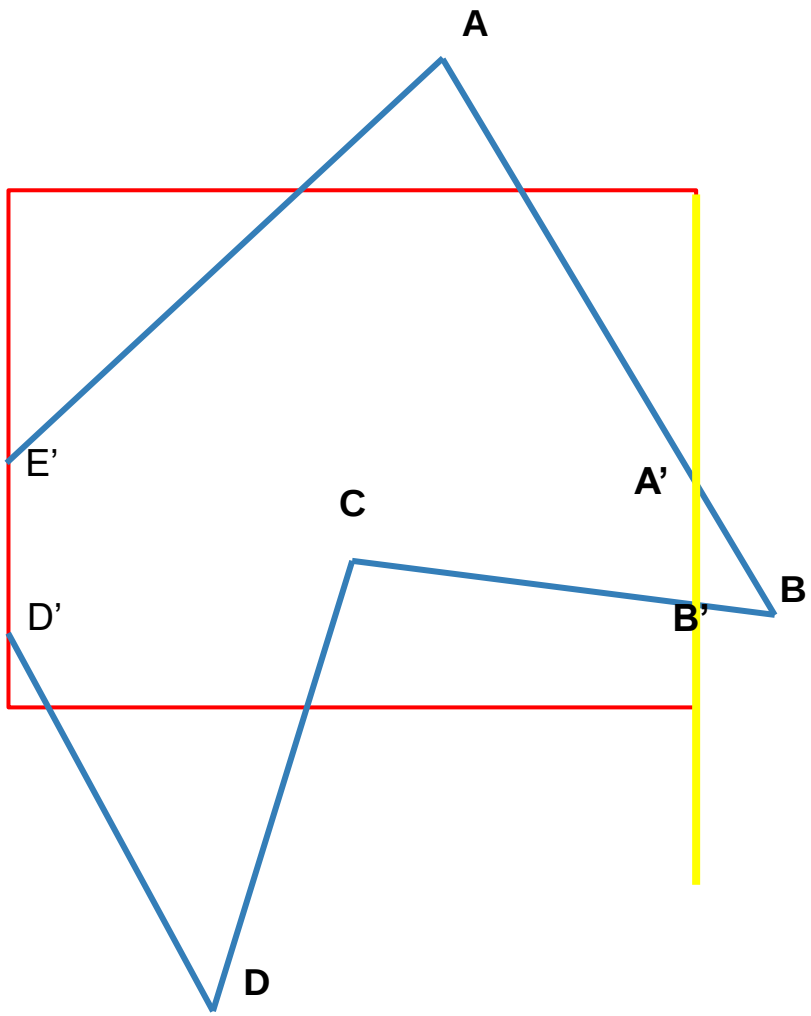


Clip the Polygon using Sutherland Hodgemen Algorithm:

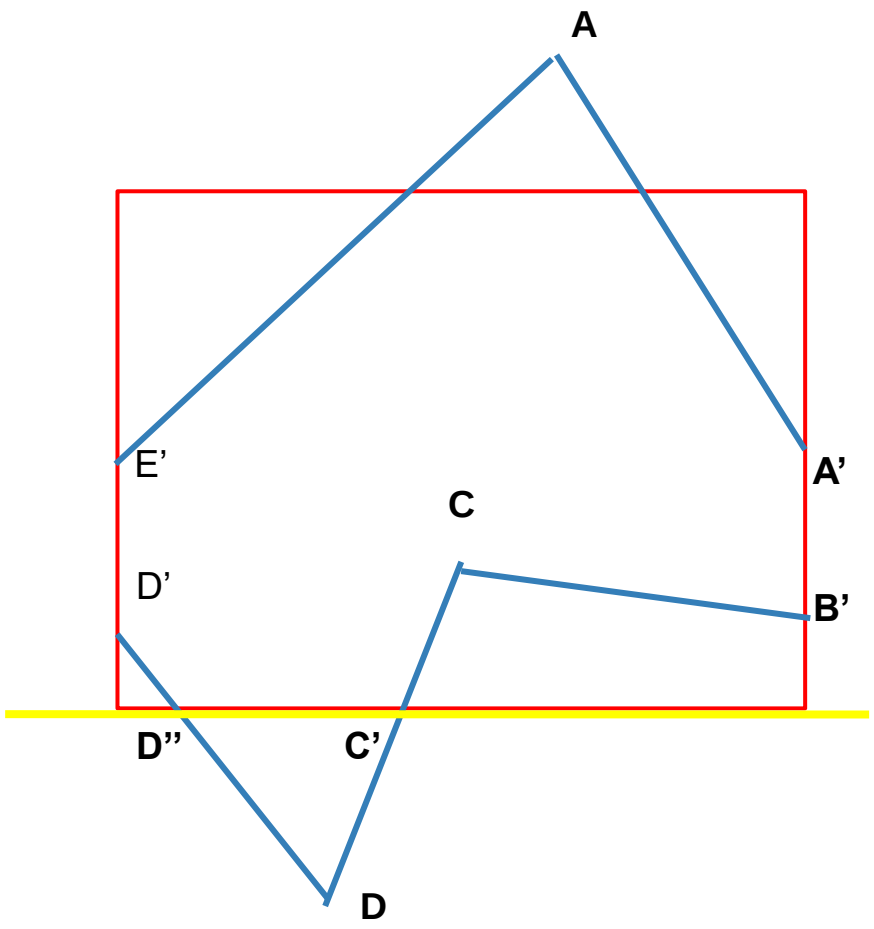




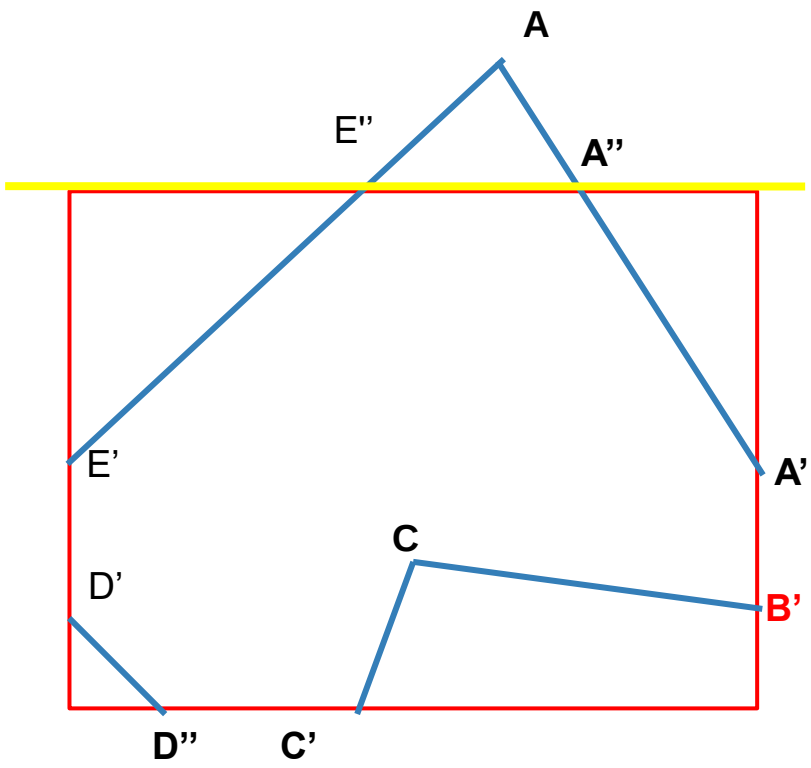
VERTEX	CASE	OUTPUT
AB	in➔in	B
BC	in➔in	C
CD	in➔in	D
DE	in➔out	D’
EA	out➔in	E’A



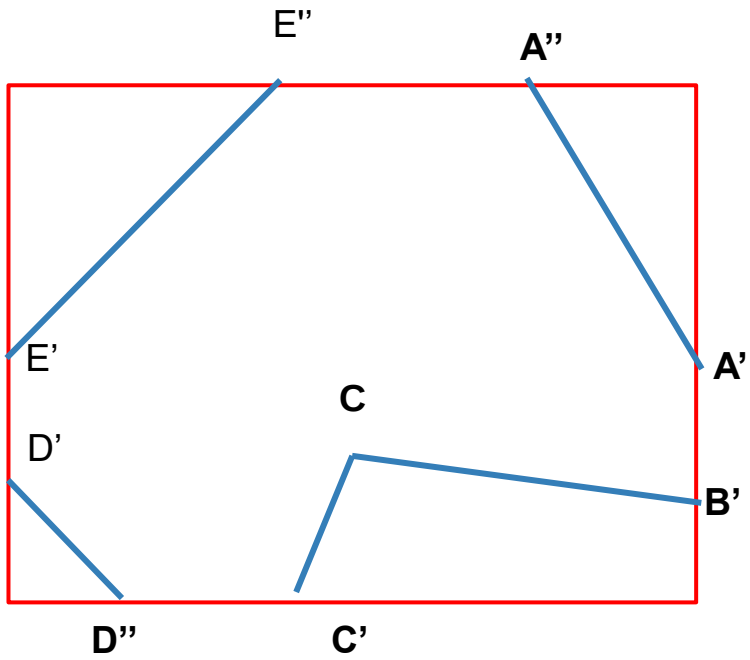
VERTEX	CASE	OUTPUT
AB	in➔out	A’
BC	out➔in	B’C
CD	in➔in	D
DD’	in➔in	D’
D’E’	in➔in	E’
E’A	in➔in	A



VERTEX	CASE	OUTPUT
AA’	in➔in	A’
A’B’	in➔in	B’
B’C	in➔in	C
CD	in➔out	C’
DD’	out➔in	D” D’
D’E’	in➔in	E’
E’A	in➔in	A



VERTEX	CASE	OUTPUT
AA'	out➔in	A'' A'
A'B'	in➔in	B'
B'C	in➔in	C
CC'	in➔in	C'
C'D''	in➔in	D''
D''D'	in➔in	D'
D'E'	in➔in	E'
E'A	in➔out	E''



VERTEX	CASE	OUTPUT
AA'	out➔in	A'' A'
A'B'	in➔in	B'
B'C	in➔in	C
CC'	in➔in	C'
C'D''	in➔in	D''
D''D'	in➔in	D'
D'E'	in➔in	E'
E'A	in➔out	E''