**COMPUTER GRAPHICS ASSIGNMENT-3**

**CLIPPING**

**Name-**ABHISHEK TIBREWAL

**Id-**2016UCP1103

**Batch-**A(1,2)

* **COHEN-SUTHERLAND LINE CLIPPING:**

**CODE:-**

**from graphics import \***

**def draw\_slope\_less(x0,y0,x1,y1,color): #for line having slope between -1 and 1**

**dy=y1-y0**

**dx=x1-x0**

**yi=1**

**if(dy<0): #if slope is between -1 and 0**

**yi=-1**

**dy=-dy**

**dne=2\*(dy-dx)**

**de=2\*dy**

**d=2\*dy-dx**

**y=y0**

**r=1**

**if x1<x0:**

**r=-1**

**for x in range(x0,x1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**#window.plot(x,y,color)**

**#window.plot(x,y,color)**

**if(d>0):**

**y=y+yi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_slope\_more(x0,y0,x1,y1,color): #for line having slope greater than 1 or less than -1**

**dy=y1-y0**

**dx=x1-x0**

**xi=1**

**if(dx<0): #if slope is less than -1**

**xi=-1**

**dx=-dx**

**dne=2\*(dx-dy)**

**de=2\*dx**

**d=2\*dx-dy**

**x=x0**

**r=1**

**if y1<y0:**

**r=-1**

**for y in range(y0,y1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**#window.plot(x,y,color)**

**#window.plot(x,y,color)**

**if(d>0):**

**x=x+xi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_line(x0,y0,x1,y1,color):**

**if abs(y1-y0) < abs(x1-x0): #if |slope|<1**

**if x0>=x1:**

**draw\_slope\_less(x1,y1,x0,y0,color) #here we interchange starting and end point of line because we have to draw from bottom to top**

**else:**

**draw\_slope\_less(x0,y0,x1,y1,color)**

**else: #if |slope|>1**

**if y0>=y1:**

**draw\_slope\_more(x1,y1,x0,y0,color)**

**else:**

**draw\_slope\_more(x0,y0,x1,y1,color)**

**INSIDE = 0 #0000**

**LEFT = 1 #0001**

**RIGHT = 2 #0010**

**BOTTOM = 4 #0100**

**TOP = 8 #1000**

**def computeCode(x, y):**

**code = INSIDE**

**if y < y\_min: # below the rectangle**

**code |= BOTTOM**

**elif y > y\_max: # above the rectangle**

**code |= TOP**

**if x < x\_min: # to the left of rectangle**

**code |= LEFT**

**elif x > x\_max: # to the right of rectangle**

**code |= RIGHT**

**return code**

**def cohenSutherlandClip(x1, y1, x2, y2):**

**code1 = computeCode(x1, y1)**

**code2 = computeCode(x2, y2)**

**accept = False**

**while True:**

**if code1 == 0 and code2 == 0:**

**accept = True**

**break**

**elif (code1 & code2) != 0:**

**break**

**else:**

**x = 1.0**

**y = 1.0**

**if code1 != 0:**

**code\_out = code1**

**else:**

**code\_out = code2**

**if code\_out & TOP:**

**x = x1 + (x2 - x1) \* (y\_max - y1) / (y2 - y1)**

**y = y\_max**

**elif code\_out & BOTTOM:**

**x = x1 + (x2 - x1) \* (y\_min - y1) / (y2 - y1)**

**y = y\_min**

**elif code\_out & RIGHT:**

**y = y1 + (y2 - y1) \*(x\_max - x1) / (x2 - x1)**

**x = x\_max**

**elif code\_out & LEFT:**

**y = y1 + (y2 - y1) \* (x\_min - x1) / (x2 - x1)**

**x = x\_min**

**if code\_out == code1:**

**x1 = x**

**y1 = y**

**code1 = computeCode(x1,y1)**

**else:**

**x2 = x**

**y2 = y**

**code2 = computeCode(x2, y2)**

**if accept:**

**draw\_line(int(x1),int(y1),int(x2),int(y2),"Red")**

**print("Enter diagonal points of rectangle:")**

**x\_min=int(input("Enter x\_min:"))**

**y\_min=int(input("Enter y\_min:"))**

**x\_max=int(input("Enter x\_max:"))**

**y\_max=int(input("Enter y\_max:"))**

**print("Enter starting and ending points of line:")**

**a0=int(input("Enter x0:"))**

**b0=int(input("Enter y0:"))**

**a1=int(input("Enter x1:"))**

**b1=int(input("Enter y1:"))**

**po=[]**

**po.append([x\_min,y\_min])**

**po.append([x\_max,y\_min])**

**po.append([x\_max,y\_max])**

**po.append([x\_min,y\_max])**

**window=GraphWin("2016UCP1103\_CLIPPING-1",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("white")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('Black')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('Black')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**for i in range(1,4):**

**x1=po[i-1][0]**

**y1=po[i-1][1]**

**x2=po[i][0]**

**y2=po[i][1]**

**draw\_line(x1,y1,x2,y2,"Black")**

**x1=po[0][0]**

**y1=po[0][1]**

**x2=po[3][0]**

**y2=po[3][1]**

**draw\_line(x1,y1,x2,y2,"Black")**

**msg =Text(Point(x\_min,y\_min),"("+str(x\_min)+","+str(y\_min)+")")**

**msg.draw(window)**

**msg =Text(Point(x\_max,y\_max),"("+str(x\_max)+","+str(y\_max)+")")**

**msg.draw(window)**

**draw\_line(a0,b0,a1,b1,"GREEN")**

**msg =Text(Point(a0,b0),"("+str(a0)+","+str(b0)+")")**

**msg.draw(window)**

**msg =Text(Point(a1,b1),"("+str(a1)+","+str(b1)+")")**

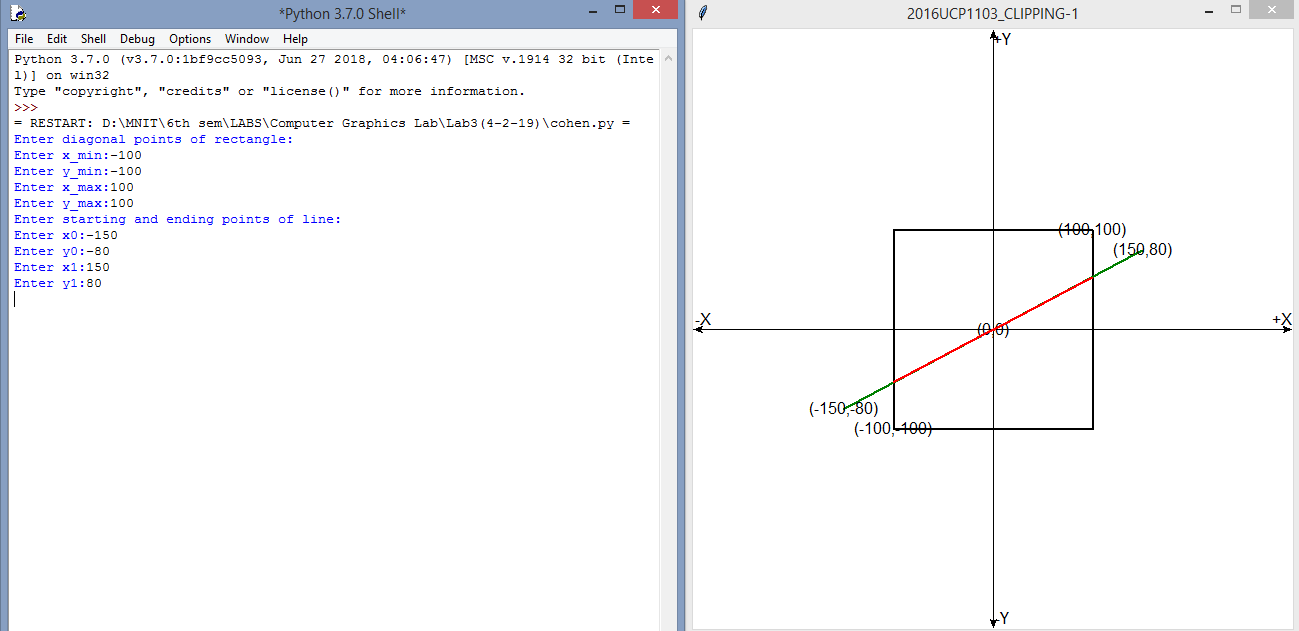
**msg.draw(window)**

**cohenSutherlandClip(a0,b0,a1,b1)**

**window.getMouse()**

**window.close()**

**OUTPUT:-**

****

* **PARAMETRIC LINE CLIPPING:**

**CODE:-**

**from graphics import \***

**def draw\_slope\_less(x0,y0,x1,y1,color): #for line having slope between -1 and 1**

**dy=y1-y0**

**dx=x1-x0**

**yi=1**

**if(dy<0): #if slope is between -1 and 0**

**yi=-1**

**dy=-dy**

**dne=2\*(dy-dx)**

**de=2\*dy**

**d=2\*dy-dx**

**y=y0**

**r=1**

**if x1<x0:**

**r=-1**

**for x in range(x0,x1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**#window.plot(x,y,color)**

**#window.plot(x,y,color)**

**if(d>0):**

**y=y+yi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_slope\_more(x0,y0,x1,y1,color): #for line having slope greater than 1 or less than -1**

**dy=y1-y0**

**dx=x1-x0**

**xi=1**

**if(dx<0): #if slope is less than -1**

**xi=-1**

**dx=-dx**

**dne=2\*(dx-dy)**

**de=2\*dx**

**d=2\*dx-dy**

**x=x0**

**r=1**

**if y1<y0:**

**r=-1**

**for y in range(y0,y1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**#window.plot(x,y,color)**

**#window.plot(x,y,color)**

**if(d>0):**

**x=x+xi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_line(x0,y0,x1,y1,color):**

**if abs(y1-y0) < abs(x1-x0): #if |slope|<1**

**if x0>=x1:**

**draw\_slope\_less(x1,y1,x0,y0,color) #here we interchange starting and end point of line because we have to draw from bottom to top**

**else:**

**draw\_slope\_less(x0,y0,x1,y1,color)**

**else: #if |slope|>1**

**if y0>=y1:**

**draw\_slope\_more(x1,y1,x0,y0,color)**

**else:**

**draw\_slope\_more(x0,y0,x1,y1,color)**

**#function for parametric line clipping**

**def parametricClip(a0,b0,a1,b1):**

**n\_left=[-1,0]**

**n\_right=[1,0]**

**n\_top=[0,1]**

**n\_bottom=[0,-1]**

**te=0.0**

**tl=1.0**

**d=[a1-a0,b1-b0]**

**ndl=n\_left[0]\*d[0]**

**f1=0**

**if a1!=a0:**

**t\_left=-1\*(-1\*(a0-x\_min))/(-1\*(a1-a0))**

**f1=1**

**if ndl<0 and f1==1:**

**te=max(t\_left,te)**

**elif ndl>=0 and f1==1:**

**tl=min(t\_left,tl)**

**ndr=n\_right[0]\*d[0]**

**f2=0**

**if a1!=a0:**

**t\_right=-1\*(a0-x\_max)/(a1-a0)**

**f2=1**

**if ndr<0 and f2==1:**

**te=max(t\_right,te)**

**elif ndr>=0 and f2==1:**

**tl=min(t\_right,tl)**

**ndb=n\_bottom[1]\*d[1]**

**f3=0**

**if b1!=b0:**

**t\_bottom=-1\*(-1\*(b0-y\_min))/(-1\*(b1-b0))**

**f3=1**

**if ndb<0 and f3==1:**

**te=max(t\_bottom,te)**

**elif ndb>=0 and f3==1:**

**tl=min(t\_bottom,tl)**

**ndt=n\_top[1]\*d[1]**

**f4=0**

**if b1!=b0:**

**t\_top=-1\*(b0-y\_max)/(b1-b0)**

**f4=1**

**if ndt<0 and f4==1:**

**te=max(t\_top,te)**

**elif ndt>=0 and f4==1:**

**tl=min(t\_top,tl)**

**x1=a0+(a1-a0)\*te**

**y1=b0+(b1-b0)\*te**

**x2=a0+(a1-a0)\*tl**

**y2=b0+(b1-b0)\*tl**

**draw\_line(int(x1),int(y1),int(x2),int(y2),"red")**

**#taking input**

**print("Enter diagonal points of rectangle:")**

**x\_min=int(input("Enter x\_min:"))**

**y\_min=int(input("Enter y\_min:"))**

**x\_max=int(input("Enter x\_max:"))**

**y\_max=int(input("Enter y\_max:"))**

**print("Enter starting and ending points of line:")**

**a0=int(input("Enter x0:"))**

**b0=int(input("Enter y0:"))**

**a1=int(input("Enter x1:"))**

**b1=int(input("Enter y1:"))**

**po=[]**

**po.append([x\_min,y\_min])**

**po.append([x\_max,y\_min])**

**po.append([x\_max,y\_max])**

**po.append([x\_min,y\_max])**

**window=GraphWin("2016UCP1103\_CLIPPING-2",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("white")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('Black')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('Black')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**for i in range(1,4):**

**x1=po[i-1][0]**

**y1=po[i-1][1]**

**x2=po[i][0]**

**y2=po[i][1]**

**draw\_line(x1,y1,x2,y2,"Black")**

**x1=po[0][0]**

**y1=po[0][1]**

**x2=po[3][0]**

**y2=po[3][1]**

**draw\_line(x1,y1,x2,y2,"Black")**

**msg =Text(Point(x\_min,y\_min),"("+str(x\_min)+","+str(y\_min)+")")**

**msg.draw(window)**

**msg =Text(Point(x\_max,y\_max),"("+str(x\_max)+","+str(y\_max)+")")**

**msg.draw(window)**

**draw\_line(a0,b0,a1,b1,"GREEN")**

**msg =Text(Point(a0,b0),"("+str(a0)+","+str(b0)+")")**

**msg.draw(window)**

**msg =Text(Point(a1,b1),"("+str(a1)+","+str(b1)+")")**

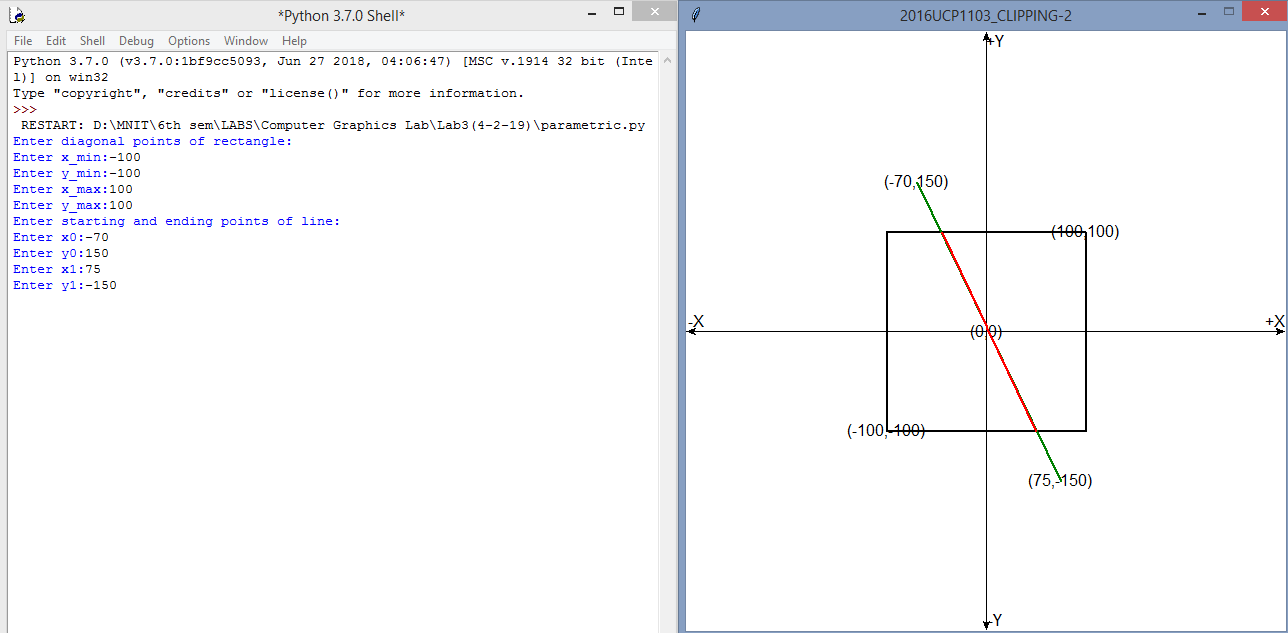
**msg.draw(window)**

**parametricClip(a0,b0,a1,b1)**

**window.getMouse()**

**window.close()**

**OUTPUT:-**

****

* **SUTHERLAND-HODGEMAN POLYGON CLIPPING:**

**CODE:-**

**from graphics import \***

**def draw\_slope\_less(x0,y0,x1,y1,color): #for line having slope between -1 and 1**

**dy=y1-y0**

**dx=x1-x0**

**yi=1**

**if(dy<0): #if slope is between -1 and 0**

**yi=-1**

**dy=-dy**

**dne=2\*(dy-dx)**

**de=2\*dy**

**d=2\*dy-dx**

**y=y0**

**r=1**

**if x1<x0:**

**r=-1**

**for x in range(x0,x1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**#window.plot(x,y,color)**

**if(d>0):**

**y=y+yi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_slope\_more(x0,y0,x1,y1,color): #for line having slope greater than 1 or less than -1**

**dy=y1-y0**

**dx=x1-x0**

**xi=1**

**if(dx<0): #if slope is less than -1**

**xi=-1**

**dx=-dx**

**dne=2\*(dx-dy)**

**de=2\*dx**

**d=2\*dx-dy**

**x=x0**

**r=1**

**if y1<y0:**

**r=-1**

**for y in range(y0,y1,r):**

**pt=Point(x,y)**

**#time.sleep(0.03)**

**pt=Point(x,y)**

**pt.setOutline(color)**

**pt.draw(window)**

**if(d>0):**

**x=x+xi**

**d=d+dne**

**else:**

**d=d+de**

**#print("("+str(x)+","+str(y)+")")**

**def draw\_line(x0,y0,x1,y1,color):**

**if abs(y1-y0) < abs(x1-x0): #if |slope|<1**

**if x0>=x1:**

**draw\_slope\_less(x1,y1,x0,y0,color) #here we interchange starting and end point of line because we have to draw from bottom to top**

**else:**

**draw\_slope\_less(x0,y0,x1,y1,color)**

**else: #if |slope|>1**

**if y0>=y1:**

**draw\_slope\_more(x1,y1,x0,y0,color)**

**else:**

**draw\_slope\_more(x0,y0,x1,y1,color)**

**#functions for polygon clipping**

**def inside(point,clip\_edge):**

**if clip\_edge[1][0]>clip\_edge[0][0]: # for bottom clipping edge**

**if point[1]>=clip\_edge[0][1]:**

**return True**

**if clip\_edge[1][0]<clip\_edge[0][0]: # for top clipping edge**

**if point[1]<=clip\_edge[0][1]:**

**return True**

**if clip\_edge[1][1]>clip\_edge[0][1]: # for right clipping edge**

**if point[0]<=clip\_edge[1][0]:**

**return True**

**if clip\_edge[1][1]<clip\_edge[0][1]: #for left clipping edge**

**if point[0]>=clip\_edge[1][0]:**

**return True**

**return False**

**def intersection(s,p,clip\_edge):**

**if clip\_edge[0][1]==clip\_edge[1][1]: #clipping edge is horizontal**

**y=clip\_edge[0][1]**

**x=s[0]+(y-s[1])\*(p[0]-s[0])//(p[1]-s[1])**

**else: #clipping edge is vertical**

**x=clip\_edge[0][0]**

**y=s[1]+(x-s[0])\*(p[1]-s[1])//(p[0]-s[0])**

**return [x,y]**

**def clippolygon(input,in\_len,clip\_edge):**

**output=[]**

**s=input[in\_len-1]**

**for i in range(0,in\_len):**

**p=input[i]**

**if inside(p,clip\_edge):**

**if inside(s,clip\_edge):**

**output.append(p)**

**else:**

**i=intersection(s,p,clip\_edge)**

**output.append(i)**

**output.append(p)**

**else:**

**if inside(s,clip\_edge):**

**i=intersection(s,p,clip\_edge)**

**output.append(i)**

**s=p**

**return output**

**#taking input**

**print("Enter diagonal points of rectangle:")**

**x\_min=int(input("Enter x\_min:"))**

**y\_min=int(input("Enter y\_min:"))**

**x\_max=int(input("Enter x\_max:"))**

**y\_max=int(input("Enter y\_max:"))**

**po=[]**

**po.append([x\_min,y\_min])**

**po.append([x\_max,y\_min])**

**po.append([x\_max,y\_max])**

**po.append([x\_min,y\_max])**

**print("Enter number of vertices in polygon:")**

**n=int(input())**

**poly=[]**

**for i in range(0,n):**

**a0=int(input("Enter x coordinate:"))**

**b0=int(input("Enter y coordinate:"))**

**poly.append([a0,b0])**

**window=GraphWin("2016UCP1103\_CLIPPING-3",600,600) #for viewport(device coordinates)**

**window.setCoords(-300,-300,300,300) #for window(user coordinates)**

**window.setBackground("white")**

**#drwing user coordinate system**

**X=Line(Point(-300,0),Point(300,0)) #for drawing X-axis**

**X.setArrow('both')**

**X.setOutline('Black')**

**X.draw(window)**

**msg=Text(Point(290,10), "+X")**

**msg.draw(window)**

**msg=Text(Point(-290,10), "-X")**

**msg.draw(window)**

**Y=Line(Point(0,-300),Point(0,300)) #for drawing Y-axis**

**Y.setArrow('both')**

**Y.setOutline('Black')**

**Y.draw(window)**

**msg=Text(Point(10,290), "+Y")**

**msg.draw(window)**

**msg=Text(Point(10,-290), "-Y")**

**msg.draw(window)**

**msg=Text(Point(0,0), "(0,0)") #for origin**

**msg.draw(window)**

**for i in range(1,4):**

**x1=po[i-1][0]**

**y1=po[i-1][1]**

**x2=po[i][0]**

**y2=po[i][1]**

**draw\_line(x1,y1,x2,y2,"yellow")**

**x1=po[0][0]**

**y1=po[0][1]**

**x2=po[3][0]**

**y2=po[3][1]**

**draw\_line(x1,y1,x2,y2,"yellow")**

**for i in range(0,n-1):**

**x1=poly[i][0]**

**y1=poly[i][1]**

**x2=poly[i+1][0]**

**y2=poly[i+1][1]**

**draw\_line(x1,y1,x2,y2,"Green")**

**x1=poly[n-1][0]**

**y1=poly[n-1][1]**

**x2=poly[0][0]**

**y2=poly[0][1]**

**draw\_line(x1,y1,x2,y2,"Green")**

**msg =Text(Point(x\_min,y\_min),"("+str(x\_min)+","+str(y\_min)+")")**

**msg.draw(window)**

**msg =Text(Point(x\_max,y\_max),"("+str(x\_max)+","+str(y\_max)+")")**

**msg.draw(window)**

**clip\_edge1=[[x\_min,y\_min],[x\_max,y\_min]]**

**clip\_edge2=[[x\_max,y\_min],[x\_max,y\_max]]**

**clip\_edge3=[[x\_max,y\_max],[x\_min,y\_max]]**

**clip\_edge4=[[x\_min,y\_max],[x\_min,y\_min]]**

**poly=clippolygon(poly,len(poly),clip\_edge1)**

**poly=clippolygon(poly,len(poly),clip\_edge2)**

**poly=clippolygon(poly,len(poly),clip\_edge3)**

**poly=clippolygon(poly,len(poly),clip\_edge4)**

**ver=len(poly)**

**for i in range(0,ver-1):**

**x1=poly[i][0]**

**y1=poly[i][1]**

**x2=poly[i+1][0]**

**y2=poly[i+1][1]**

**draw\_line(x1,y1,x2,y2,"RED")**

**x1=poly[ver-1][0]**

**y1=poly[ver-1][1]**

**x2=poly[0][0]**

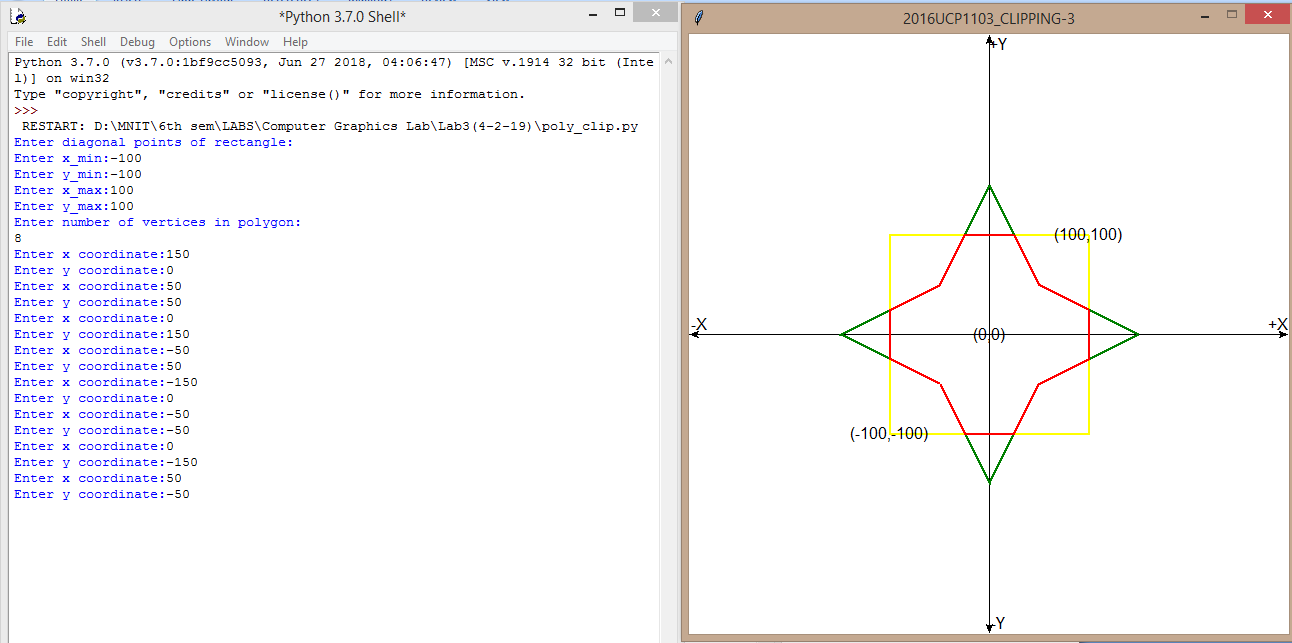
**y2=poly[0][1]**

**draw\_line(x1,y1,x2,y2,"RED")**

**window.getMouse()**

**window.close()**

**OUTPUT:-**

****