```
############################################################################
####
#graphics
############################################################################
####
readlib('gdi')
wh := gdi.open(640, 480) #window handler
initcenter := ["x" ~ 320,"y" ~ 240]
iter := 9 #max iterations
inititer := iter #... reset by functions to start spawning again

#golden ratio spiral, grow from outside in
grs_in := proc(center,scale,rotate) #golden ratio
        initsize := 200
        #gdi.rectangle(wh, center["x"] - initsize * scale,
        #                       center["y"] - initsize * scale,
        #                       center["x"] + initsize * scale,
        #                       center["y"] + initsize * scale)
        gdi.arc(wh, center["x"], center["y"],
                        initsize * scale,
                        initsize * scale,
                        rotate,
                        rotate + 180)
        iter := iter - 1
        if iter <= 0 then
                iter := inititer
                return
        fi;
        os.wait(1)
        x := center["x"] - (((initsize * scale) / 2) * cos((rotate % 360) *
radians))
        y := center["y"] - (((initsize * scale) / 2) * sin((rotate % 360) *
radians))
        gr(["x" ~ x, "y" ~ y],
          scale / 2,
          rotate + 180
          )
end

#golden ratio spiral - grow from inside out
grs_out := proc(center,scale,rotate) #golden ratio
        initsize := 200
        #gdi.rectangle(wh, center["x"] - initsize * scale,
        #                       center["y"] - initsize * scale,
        #                       center["x"] + initsize * scale,
        #                       center["y"] + initsize * scale)
        gdi.arc(wh, center["x"], center["y"],
                        initsize * scale,
                        initsize * scale,
                        rotate,
                        rotate + 180)
        iter := iter - 1
        if iter <= 0 then
                iter := inititer
                return
        fi;
        x := center["x"] - (((initsize * scale) ) * cos((rotate % 360) *
radians))
        y := center["y"] - (((initsize * scale) ) * sin((rotate % 360) *
radians))
        #os.wait(1)
        grs_out(["x" ~ x, "y" ~ y],
          scale * 2,
          rotate + 180
          )
end

grs_out_multi := proc(n, step) #n - how many spiralss, step - degrees
between each spiral
        stop := n * step
        current := 0
        for current to stop by step while current <= stop do
                grs_out(initcenter,0.00390625, current)
                current := current + step
                os.wait(1)
        od
end

######################################################################
#Start Process
######################################################################
#for n, try something fibonacci-ish to see something natural!
grs_out_multi(100,152)
```