

```
> u := [1,1];  
u := [1, 1] (1)
```

```
> v := [2,3];  
v := [2, 3] (2)
```

```
> u+v;  
[3, 4] (3)
```

```
> u-v;  
[-1, -2] (4)
```

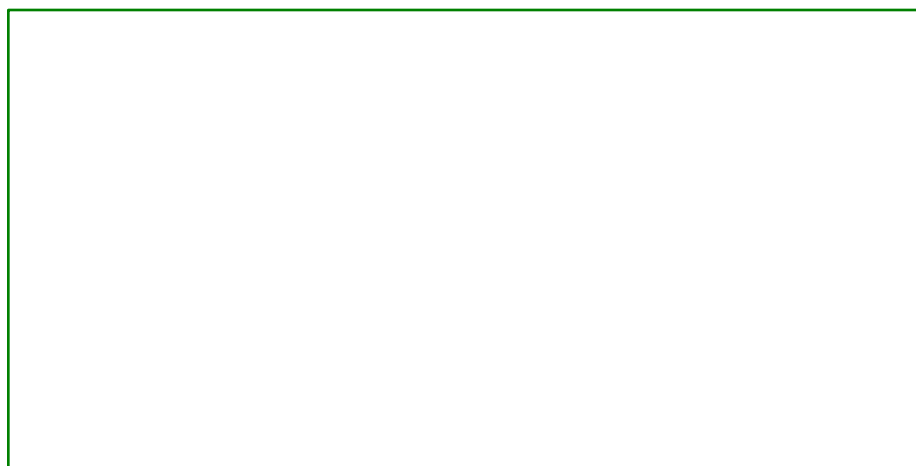
```
> alpha := 0.3;  
 $\alpha := 0.3$  (5)
```

```
> alpha*u;  
[0.3, 0.3] (6)
```

```
> 0.5*(u+v);  
[1.5, 2.0] (7)
```

```
> P := plot( [[0,0],[0,1],[1,1],[1,0]], style=line, axes=none );  
P := PLOT(...)
```

```
> PLOT( CURVES( [[0,0],[0,1],[1,1],[1,0]], COLOR(RGB,0,.5,0) ),  
AXESSTYLE(NONE) );
```



```
> PLOT( POLYGONS( [[0,0],[0,1],[1,1],[1,0]], COLOR(RGB,0,.5,0) ),  
AXESSTYLE(NONE) );
```



```
> line := proc(a,b) CURVES([a,b],COLOR(RGB,0,0.5,0),THICKNESS(2))
end;
line:=proc(a,b) CURVES([a,b],COLOR(RGB,0,0.5,0),THICKNESS(2)) end proc (9)
```

```
> poly := proc(a,p,q,r,b)
POLYGONS([a,p,q,r,b],COLOR(RGB,0,0.5,0),STYLE(PATCHNOGRID))
end;
poly:=proc(a,p,q,r,b) (10)
POLYGONS([a,p,q,r,b],COLOR(RGB,0,0.5,0),STYLE(PATCHNOGRID))
end proc
```

```
> PLOT( line([0,0],[0,1]), line( [1,1],[1,0] ), AXESSTYLE(NONE) )
;
```



```
> broc := proc(a,b,n::nonnegint) local u,v,p,q,r;
if n=0 then return line(a,b) fi;
u := b-a;
v := [-u[2],u[1]];
p := a+v;
r := b+v;
```

```

    q := 0.5*(p+r)+alpha*v;
    line(a,p), broc(p,q,n-1), broc(q,r,n-1), line(r,b);
end:
> alpha := 0.3;

```

$\alpha := 0.3$

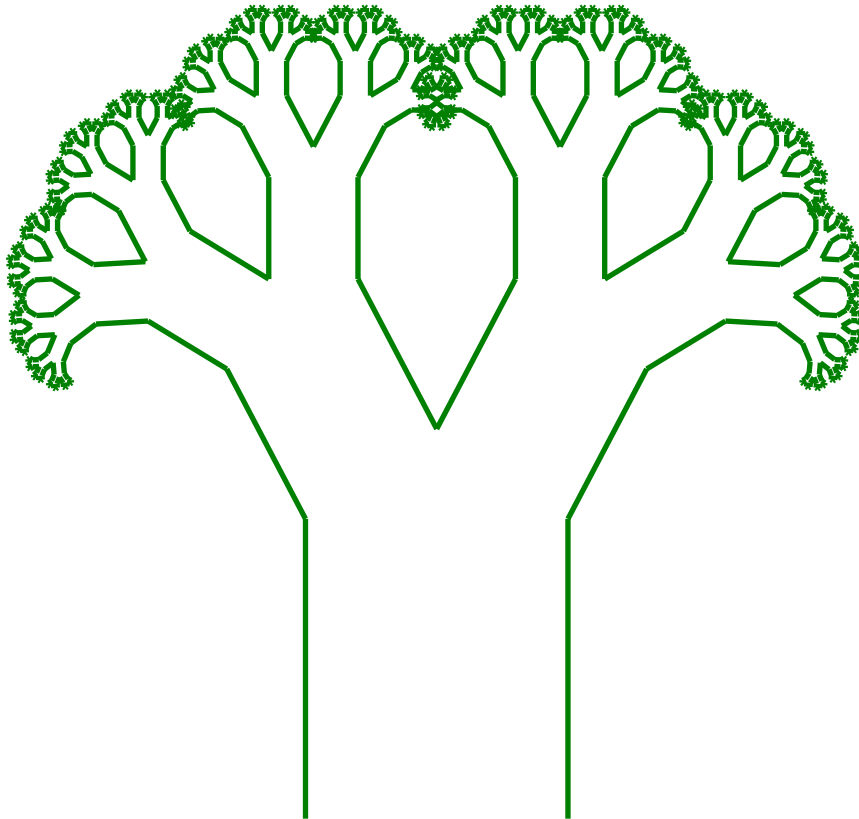
(11)

This is the broccoli fractal using lines for the boundary

```

> PLOT( broc([0,0],[1,0],9), AXESSTYLE(NONE) );

```



```

> broc := proc(a,b,n::nonnegint) local u,v,p,q,r;
    if n=0 then return NULL; fi;
    u := b-a;
    v := [-u[2],u[1]];
    p := a+v;
    r := b+v;
    q := 0.5*(p+r)+alpha*v;
    poly(a,p,q,r,b), broc(p,q,n-1), broc(q,r,n-1);
end:

```

```

> alpha := 0.35;

```

$\alpha := 0.35$

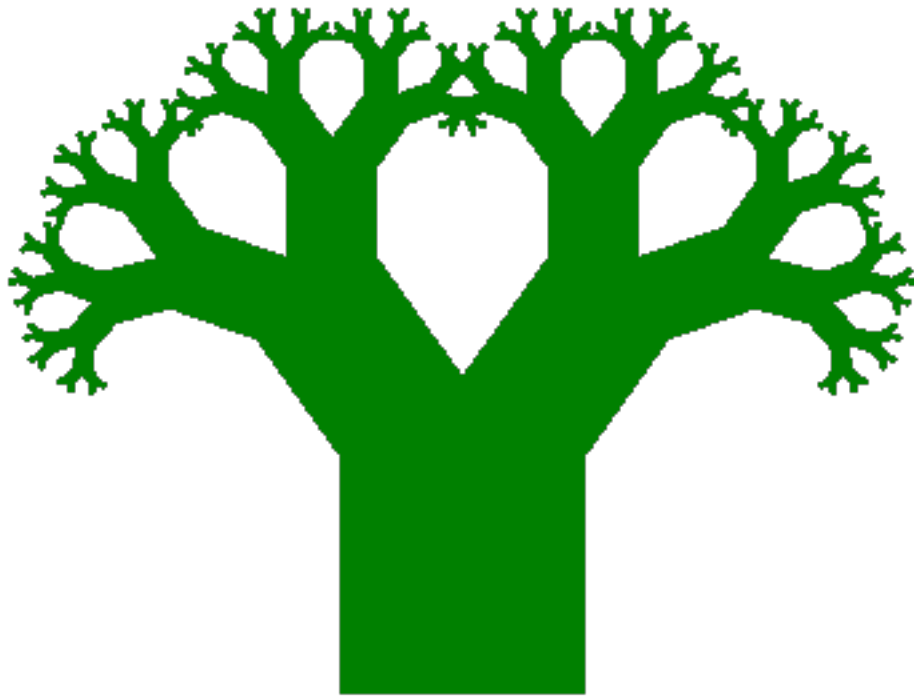
(12)

This version uses polygons to make it solid.

```

> PLOT( broc([0,0],[1,0],8), AXESSTYLE(NONE) );

```



```

> U := rand(-1000..1000);
U:=proc()
  proc() option builtin = RandNumberInterface; end proc(6, 2001, 11) - 1000
end proc

```

(13)

```

> beta := proc() U()/10000.0; end;
      β:=proc() U()/10000.0 end proc

```

(14)

```

> beta();
      -0.014000000000

```

(15)

This final version places the key vertex q randomly to make the fractal asymmetric

```

> broc := proc(a,b,n::nonnegint) local u,v,p,q,r;
  if n=0 then return NULL; fi;
  u := b-a;
  v := [-u[2],u[1]];
  p := a+v;
  r := b+v;
  q := 0.5*(p+r)+alpha*v;
  q := q + beta()*u;
  poly(a,p,q,r,b), broc(p,q,n-1), broc(q,r,n-1);
end:
> alpha := 0.35;
      α:=0.35
> PLOT( broc([0,0],[1,0],9), AXESSTYLE(NONE) );

```

(16)

