Koch Flocke

#F--F--F

#F → F+F--F+F

function main() {

turtle->setSpeed(100000)

var step := 500

var iteration := 4

center(step)

for(var i < 3) {

koch(step,iteration)

turtle->rightTurn(120)

}

}

function koch(step,iteration) {

if(iteration > 0) {

koch(step/3,iteration-1)

turtle->leftTurn(60)

koch(step/3,iteration-1)

turtle->rightTurn(60)

turtle->rightTurn(60)

koch(step/3,iteration-1)

turtle->leftTurn(60)

koch(step/3,iteration-1)

}else {

turtle->forward(step)

}

}

function center(step) {

turtle->penUp()

var up := step / 4

turtle->forward(up)

turtle->leftTurn(90)

var right := step / 2

turtle->forward(right)

turtle->rightTurn(180)

turtle->penDown()

}

Penta Plexity

function main() {

# F++F++F++F++F

# F → F++F++F|F-F++F

turtle->setSpeed(3000)

var step := 10

var iteration := 3

for(var i < 4) {

pentaplexity(step, iteration)

turtle->rightTurn(36)

turtle->rightTurn(36)

}

pentaplexity(step,iteration)

}

function pentaplexity(step,iteration) {

if(iteration>0) {

pentaplexity(step, iteration-1)

turtle->rightTurn(36)

turtle->rightTurn(36)

pentaplexity(step, iteration-1)

turtle->rightTurn(36)

turtle->rightTurn(36)

pentaplexity(step, iteration-1)

turtle->rightTurn(180)

pentaplexity(step,iteration-1)

turtle->leftTurn(36)

pentaplexity(step, iteration-1)

turtle->rightTurn(36)

turtle->rightTurn(36)

pentaplexity(step, iteration-1)

}else {

turtle->forward(step)

}

}

Peano-Kurve

function main() {

# F → F-F+F+F+F-F-F-F+F

# F

turtle->setSpeed(3000)

var iteration := 3

var step := 10

kurve(step, iteration)

}

function kurve(step,iteration) {

if(iteration>0) {

kurve(step, iteration-1)

turtle->leftTurn(90)

kurve(step, iteration-1)

turtle->rightTurn(90)

kurve(step, iteration-1)

turtle->rightTurn(90)

kurve(step, iteration-1)

turtle->rightTurn(90)

kurve(step, iteration-1)

turtle->leftTurn(90)

kurve(step, iteration-1)

turtle->leftTurn(90)

kurve(step, iteration-1)

turtle->leftTurn(90)

kurve(step, iteration-1)

turtle->rightTurn(90)

kurve(step, iteration-1)

}else {

turtle->forward(step)

}

}