* 1. Hello, world!
  2. Hello world!
  3. 3
  4. 3.0
  5. 5
  6. 5.0
  7. 23
  8. 2 + 3= 5
  9. 6
  10. 8
  11. 0.6666666666666666

1. def main():

print("This program illustrates a chaotic function")

x = eval(input("Enter a number between 0 and 1: "))

for i in range(10):

x = 3.9 \* x \* (1-x)

print(x)

main()

This program illustrates a chaotic function

Enter a number between 0 and 1: .5

0.975

0.09506250000000008

0.33549992226562525

0.8694649252590003

0.44263310911310905

0.962165255336889

0.1419727793616139

0.4750843861996143

0.9725789275369049

0.1040097132674683

1. def main():

print("This program illustrates a chaotic function")

x = eval(input("Enter a number between 0 and 1: "))

for i in range(10):

x = 2.0 \* x \* (1-x)

print(x)

main()

This program illustrates a chaotic function

Enter a number between 0 and 1: .5

0.5

0.5

0.5

0.5

0.5

0.5

0.5

0.5

0.5

0.5

With the 2.0 multiplier, the program printed out 0.5 for the 0.5 input. Also, I noticed a lot of the results were either 0.5. or started with 0.49999…

1. This program illustrates a chaotic function

Enter a number between 0 and 1: .5

0.975

0.09506250000000008

0.33549992226562525

0.8694649252590003

0.44263310911310905

0.962165255336889

0.1419727793616139

0.4750843861996143

0.9725789275369049

0.1040097132674683

0.36344760197260056

0.90227842611257

0.34387106474913476

0.8799326467519814

0.4120396173349332

0.9448255872175185

0.20330776813070028

0.6316975062388134

0.907357490716863

0.3278335115517574

1. def main():

print("This program illustrates a chaotic function")

x = eval(input("Enter a number between 0 and 1: "))

n = eval(input("How many numbers should I print? "))

for i in range(n):

x = 3.9 \* x \* (1-x)

print(x)

main()

This program illustrates a chaotic function

Enter a number between 0 and 1: .5

How many numbers should I print? 5

0.975

0.09506250000000008

0.33549992226562525

0.8694649252590003

0.44263310911310905

Enter a number between 0 and 1: .5

0.975

0.09506250000000008

0.33549992226562525

0.8694649252590003

0.44263310911310905

0.962165255336889

0.1419727793616139

0.4750843861996143

0.9725789275369049

0.1040097132674683

0.36344760197260056

0.90227842611257

0.34387106474913476

0.8799326467519814

0.4120396173349332

0.9448255872175185

0.20330776813070028

0.6316975062388134

0.907357490716863

0.3278335115517574

0.8593989309960643

0.4712463927556561

0.9717755972747083

0.10696836468275975

0.3725519211954379

0.9116522501152019

0.3141154574028559

0.8402430536114569

0.5235151914296899

0.9728434395108977

0.10303441867486574

0.360431476248473

0.8990304459934958

0.35402134236390415

0.8918919029075777

0.376040872098363

0.9150731249784758

0.3030857735903501

0.8237766710062077

0.566157802517338

0.9579302661476999

0.15716949824899712

0.516622263569707

0.9739224313798948

0.09905036323637888

0.3480336162385691

0.8849342510052466

0.3971199273718163

0.9337211935584767

0.24135511240702046

0.7141010062758585

0.7962269605354927

0.6327733926224242

0.9062477822249732

0.33135468380543404

0.8640791535699761

0.458040842749502

0.9681337735790292

0.1203180031351671

0.4127821669012588

0.9453328933992844

0.20154659482082424

0.6276097032541235

0.9114914781780388

0.3146315772087253

0.8409903365443133

0.5215298024952442

0.9731922236576114

0.10174756593286385

0.3564404951624454

0.8946236074261049

0.367661613001829

0.906697550174217

0.3299287004609318

0.8621954369850608

0.4633764151660832

0.9697689808322603

0.114336708126496

0.3949289186750406

0.9319442646898946

0.24735419358587463

0.7260633763552895

0.7756918644963023

0.6785765838171546

0.8506305744775596

0.4955269809419923

0.974921969191976

0.09535158039738144

0.33641266040110224

0.8706328110595242

0.4392621455278487

0.960612560833067

0.14756066833093617

0.49056741822138816

0.9746530029638603

0.09634765443181968

0.33955265727749473

0.874600935831819

0.4277291416083092

0.9546299998065798

nter a number between 0 and 1: .5

0.975

0.09506250000000013

0.3354999222656254

0.8694649252590004

0.442633109113109

0.962165255336889

0.14197277936161376

0.47508438619961396

0.972578927536905

0.10400971326746805

0.3634476019725998

0.9022784261125693

0.3438710647491371

0.8799326467519842

0.4120396173349247

0.9448255872175128

0.20330776813072024

0.6316975062388596

0.9073574907168157

0.3278335115519075

0.8593989309962661

0.4712463927550907

0.9717755972745815

0.10696836468322625

0.37255192119686803

0.9116522501166235

0.3141154573982912

0.8402430536048388

0.5235151914472539

0.972843439507676

0.10303441868674754

0.360431476285263

0.8990304460335468

0.3540213422392481

0.8918919027656401

0.37604087253223184

0.9150731253979755

0.30308577223219035

0.823776668920168

0.5661578077855438

0.9579302634291429

0.15716950795929147

0.5166222895357901

0.9739224280132946

0.09905037568133569

0.34803365515901563

0.8849342971391169

0.39711978885545485

0.9337210824039298

0.24135548844561772

0.7141017649069006

0.7962256936286262

0.6327763198935332

0.9062447506145676

0.3313642901326204

0.8640917896943473

0.4580049578409518

0.9681220240928331

0.12036090518099313

0.4129092149715199

0.9454192611361799

0.2012465590547083

0.6269108879410576

0.9121851434358479

0.31240328937000106

0.8377491492271373

0.5301094975657453

0.9714643308097197

0.10811340061915851

0.37605708358030737

0.9150887985305862

0.30303502840429425

0.823698719860781

0.5663546411659174

0.9578285402233951

0.15753280825196894

0.5175932680472475

0.9737928599857291

0.09952927072351882

0.34953046047177694

0.8866997789292416

0.39180679580669525

0.9293474992089048

0.25607682719993213

0.7429557945074781

0.7447926794694363

0.7412985219048781

0.7479225909734322

0.7352841166515791

0.759101399360796

0.7131792129121813

0.7977630304103573

0.6292149931112786

0.9098835936654835

0.3197822146110883

0.848334044335113

0.5017872348728597

0.974987542586886

0.09510865411035761

0.3356456922923285

Enter a number between 0 and 1: .5

0.975

0.09506250000000005

0.3354999222656252

0.8694649252590001

0.4426331091131095

0.9621652553368891

0.14197277936161345

0.47508438619961313

0.9725789275369049

0.10400971326746822

0.3634476019726003

0.9022784261125698

0.3438710647491354

0.8799326467519822

0.4120396173349308

0.9448255872175169

0.20330776813070628

0.6316975062388273

0.9073574907168487

0.32783351155180274

0.8593989309961254

0.47124639275548486

0.9717755972746699

0.10696836468290094

0.3725519211958707

0.9116522501156321

0.3141154574014746

0.8402430536094543

0.5235151914350049

0.9728434395099228

0.1030344186784613

0.3604314762596061

0.8990304460056157

0.354021342326182

0.8918919028646259

0.3760408722296562

0.9150731251054207

0.30308577317935725

0.823776670374951

0.5661578041115503

0.9579302653250372

0.1571695011874299

0.516622271427305

0.9739224303611282

0.09905036700234593

0.34803362801628135

0.8849342649658136

0.3971198854553921

0.9337211599220245

0.24135522619998495

0.7141012358451623

0.7962265771573394

0.6327742784440229

0.9062468648363093

0.33135759076161087

0.8640829774444595

0.45802998358719105

0.9681302191170033

0.12033098200291947

0.41282060341521687

0.945359035963341

0.20145578343463377

0.6273982679459497

0.9117017571650508

0.31395648627411754

0.8400124629020567

0.5241269477779822

0.9727297725245823

0.10345359245854135

0.3617286919972572

0.9004360769945214

0.34963869814082704

0.8868267677225372

0.3914237019160036

0.9290236312280631

0.25716102300675914

0.7450140018862172

0.7408757416308327

0.7487176206658055

0.7337442261643279

0.7619181832658741

0.7074555745713456

0.8071525198591643

0.6070635852222463

0.9302958160054993

0.2528974918398301

0.7368673667977174

0.7561860171306307

0.7190380260443021

0.7878871382718011

0.6517718829089243

0.8851646526772912

0.3964279422791286

0.9331640325519114

0.24323879152313

0.7178873591038395

0.7898478850967425

0.6473539936692876

0.8903185221438752

0.38084065995854255

In first few iterations, it can be seen that the output numbers are very similar. However, as more iterations are printed, the numbers differ more and eventually there seems to be no relationship between the three ouputs.

7. def main():

print("This program illustrates a chaotic function")

input\_1 = eval(input("Enter a number between 0 and 1: "))

input\_2 = eval(input("Enter another number between 0 and 1: "))

for i in range(10):

input\_1 = 3.9 \* input\_1 \* (1 - input\_1)

print(input\_1)

print("-----------------------")

for i in range(10):

input\_2 = 3.9 \* input\_2 \* (1 - input\_2)

print(input\_2)

main()