

گزارش تمرین سری اول

شبیه سازی رایانه ای در فیزیک

علیرضا رضایی

97100762

## فهرست

3	تمرین 2.1
4	نتایج: (به ترتیب مراحل)
7	تمرین 2.2
8	نتیجه
12	تمرین 2.3
13	نتیجه
16	تمرین 2.4
17	نتیجه

## تمرین 2.1

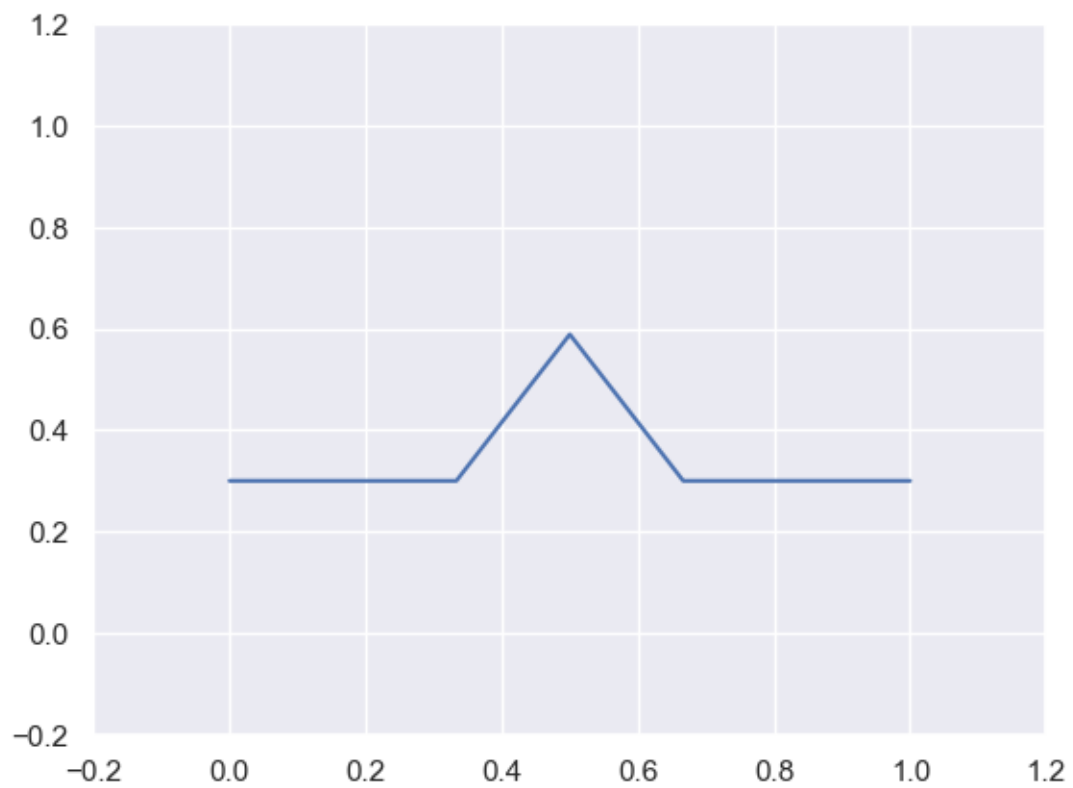
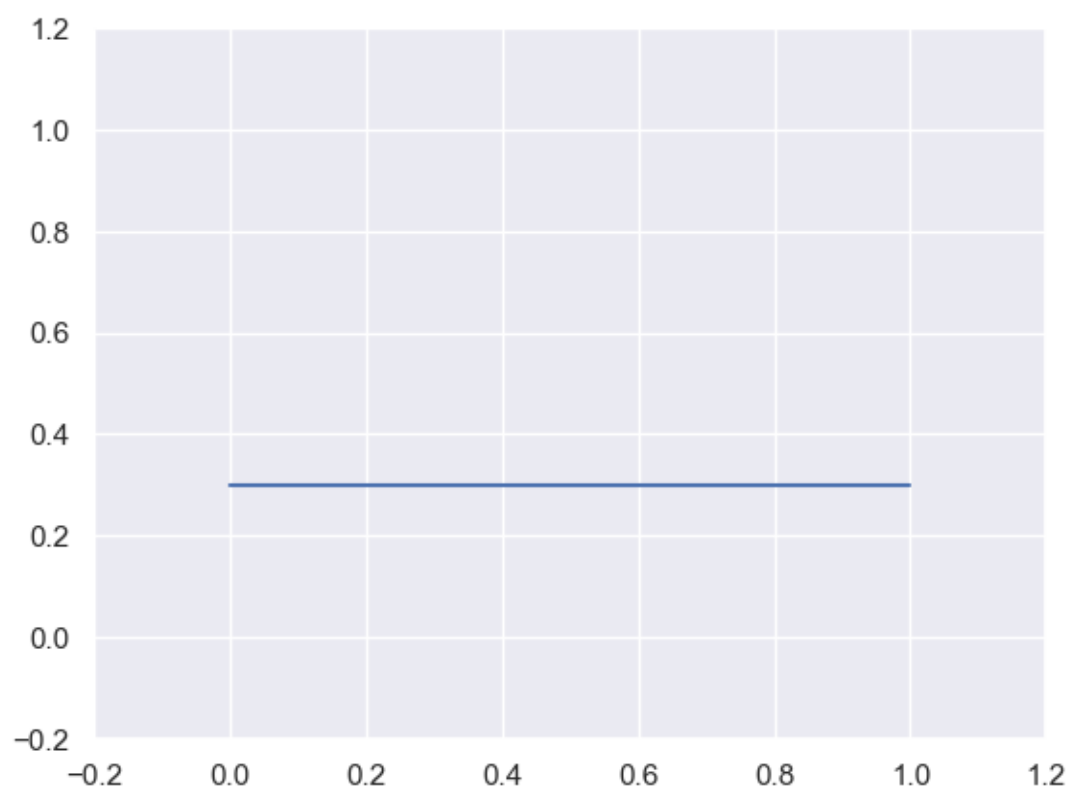
در این تمرین می‌خواهیم برف کوخ را شبیه سازی کنیم. برای این کار از یک خط شروع می‌کنیم و توابع خود متشابهی را روی آن اعمال می‌کنیم که به شرح زیر اند:

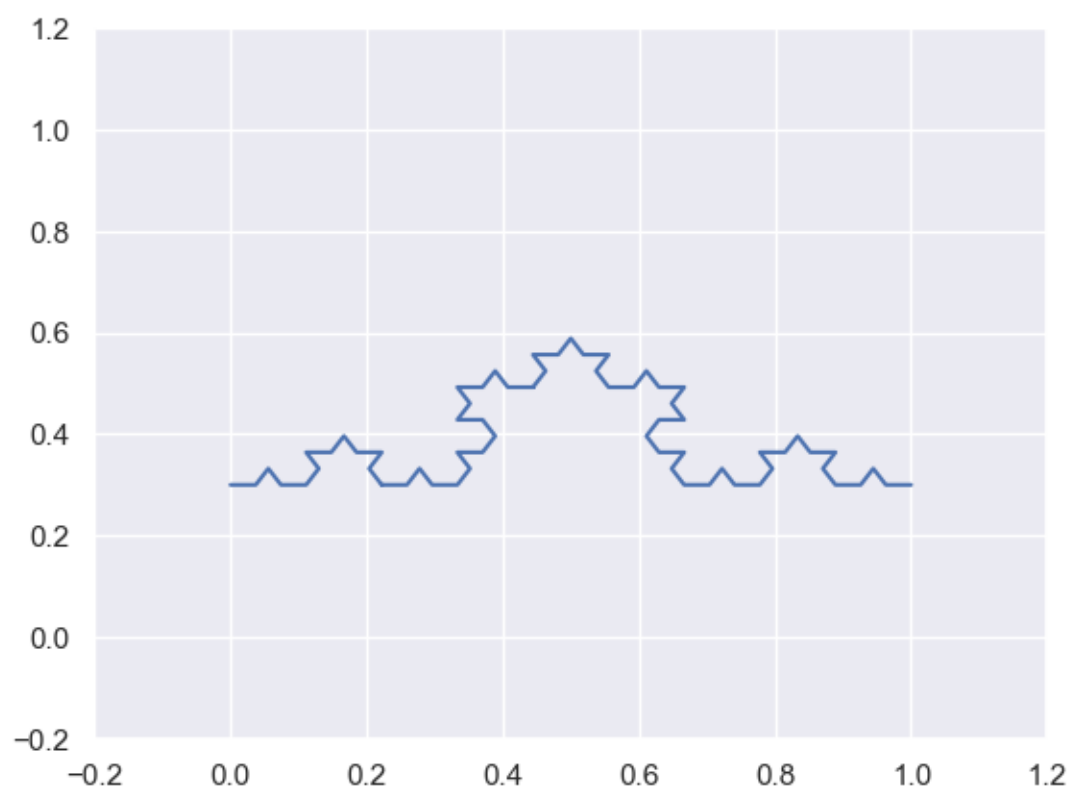
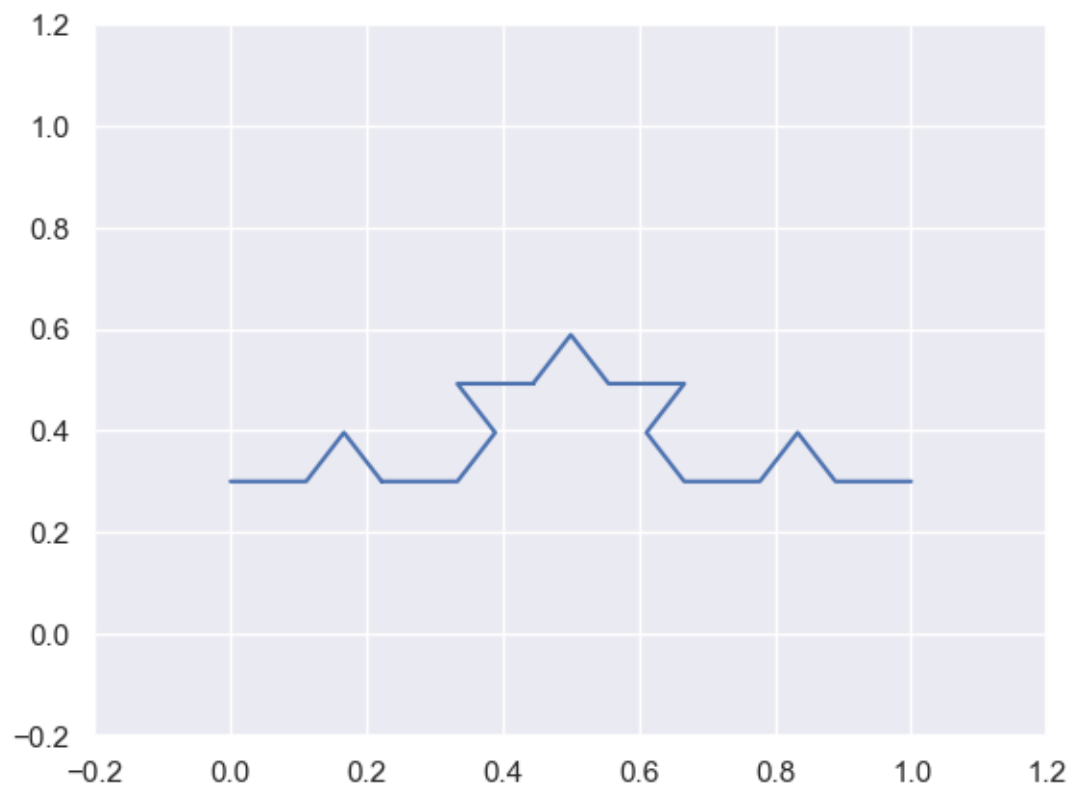
- 1- کل شکل را با ضریب  $\frac{1}{3}$  کوچک می‌کنیم و در سمت چپ قرار می‌دهیم.
- 2- کل شکل را با ضریب  $\frac{1}{3}$  کوچک می‌کنیم ، 60 درجه در خلاف جهت عقربه های ساعت دوران داده و با بردار انتقالی به اندازه ی  $\frac{1}{3}$  طول اولین شکلی که داشتیم (خط اولیه) به سمت راست انتقال می‌دهیم.
- 3- کل شکل را با ضریب  $\frac{1}{3}$  کوچک می‌کنیم ، 60 درجه در جهت عقربه های ساعت دوران داده و با بردار انتقالی نقطه ی ابتدای این خط را به نقطه ی انتهای آخرین خطی که رسم کردیم وصل می‌کنیم. (که برای یافتن این بردار فقط کافیست مختصات این دو نقطه را از هم کم کنیم.)
- 4- کل شکل را با ضریب  $\frac{1}{3}$  کوچک می‌کنیم و با بردار انتقالی به اندازه ی  $\frac{2}{3}$  طول اولین شکلی که داشتیم (خط اولیه) به سمت راست انتقال می‌دهیم.

توضیحات فنی و نحوه ی تبدیل این توابع به کد هم در فایل کد ها با کامنت گذاری توضیح داده شده است فقط به یک سری نکات کلی در این جا اشاره می‌کنیم:

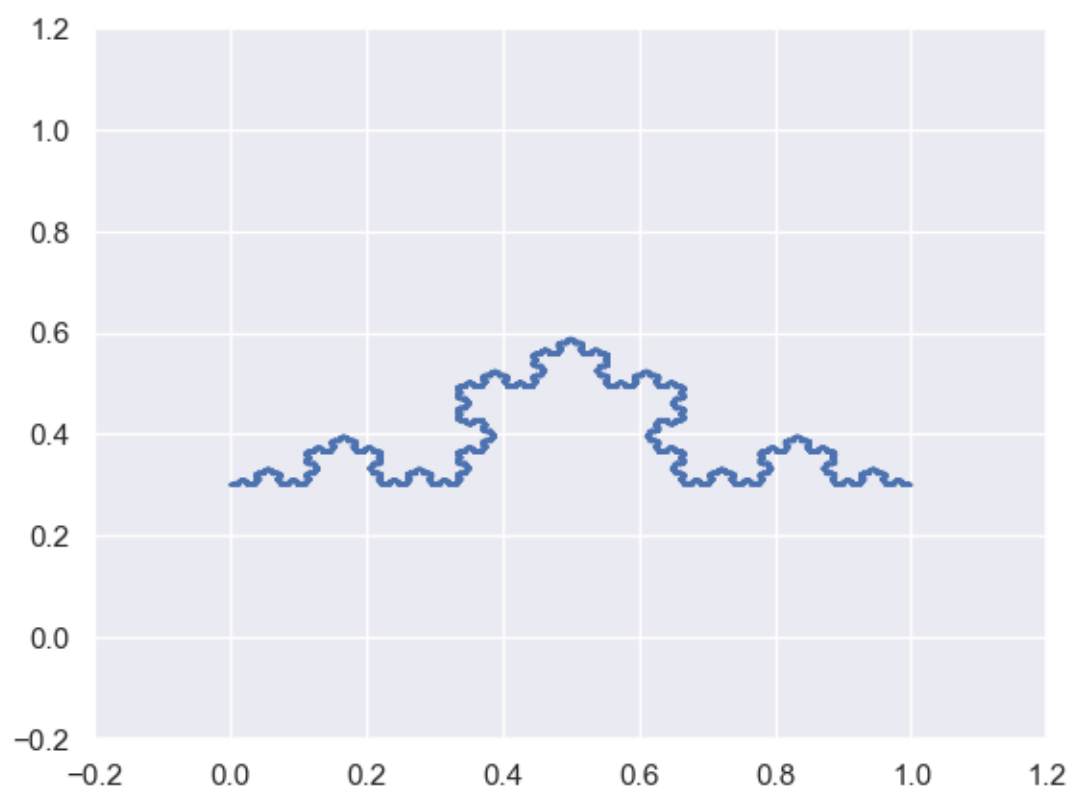
- 1- برای راحتی در نوشتن و دیباگ کردن کد از چند تابع کمکی برای دوران ، انتقال ، اسکیل کردن و ... استفاده کرده ایم.
- 2- درقسمت تابع f1 مستقیما نقاط اسکیل شده را در نقاط اصلی هم می‌ریزیم که در f های بعد مجبور نباشیم هر بار دوباره نقاط را اسکیل کنیم.

نتایج: (به ترتیب مراحل)





و ادامه می‌دهیم تا:



## تمرین 2.2

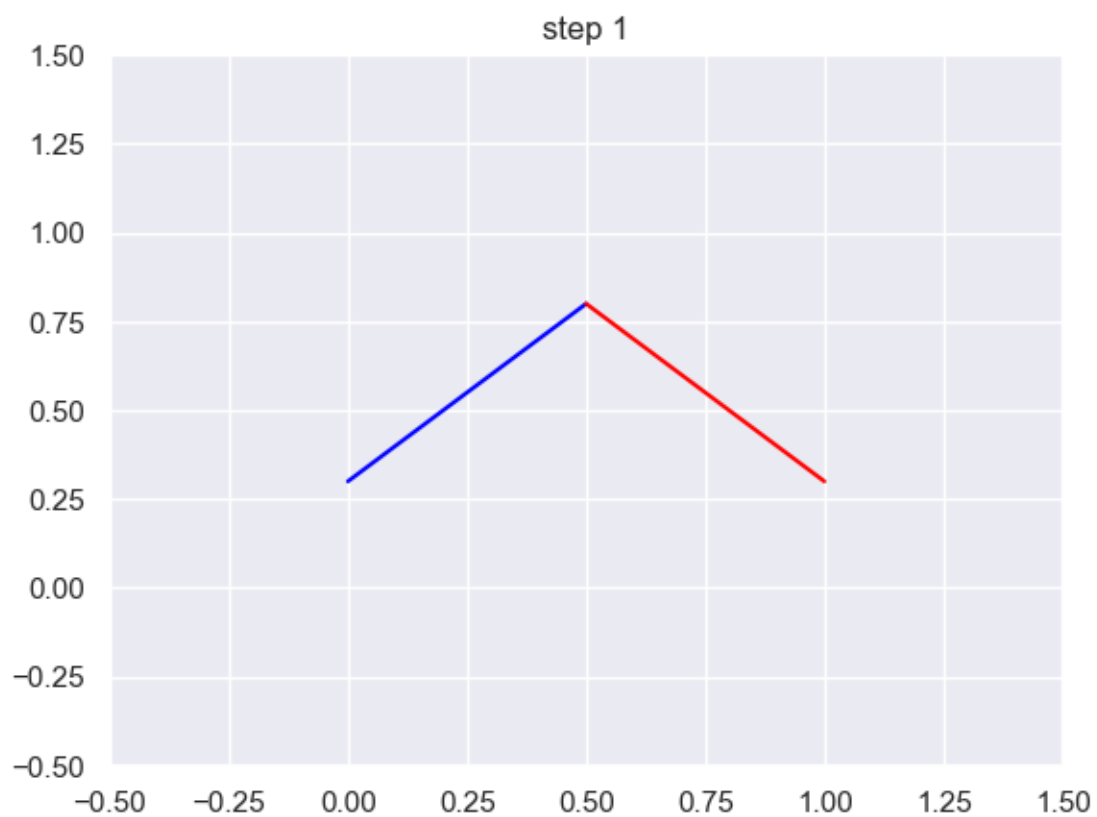
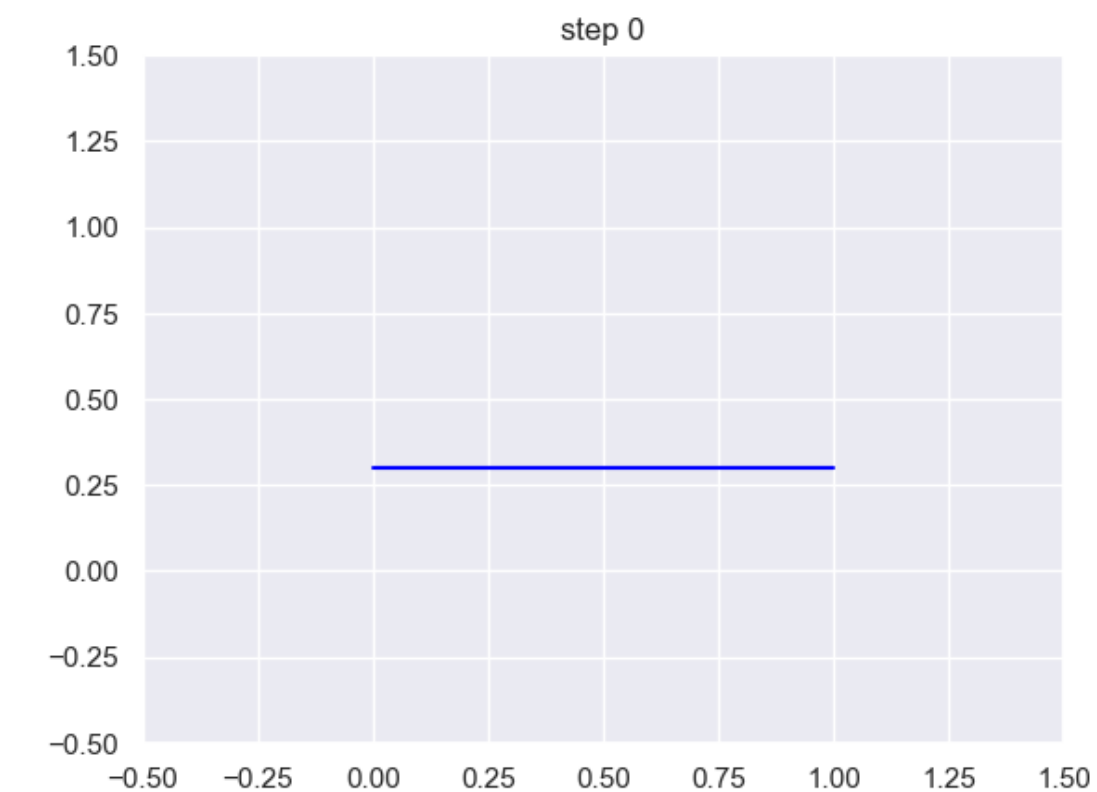
در این قسمت هم مانند قسمت قبل عمل می کنیم و از یک خط شروع میکنیم با این تفاوت که توابع خودمتشابهی که در هر مرحله اعمال میکنیم با قسمت قبل متفاوت اند.

این توابع به شرح زیر اند:

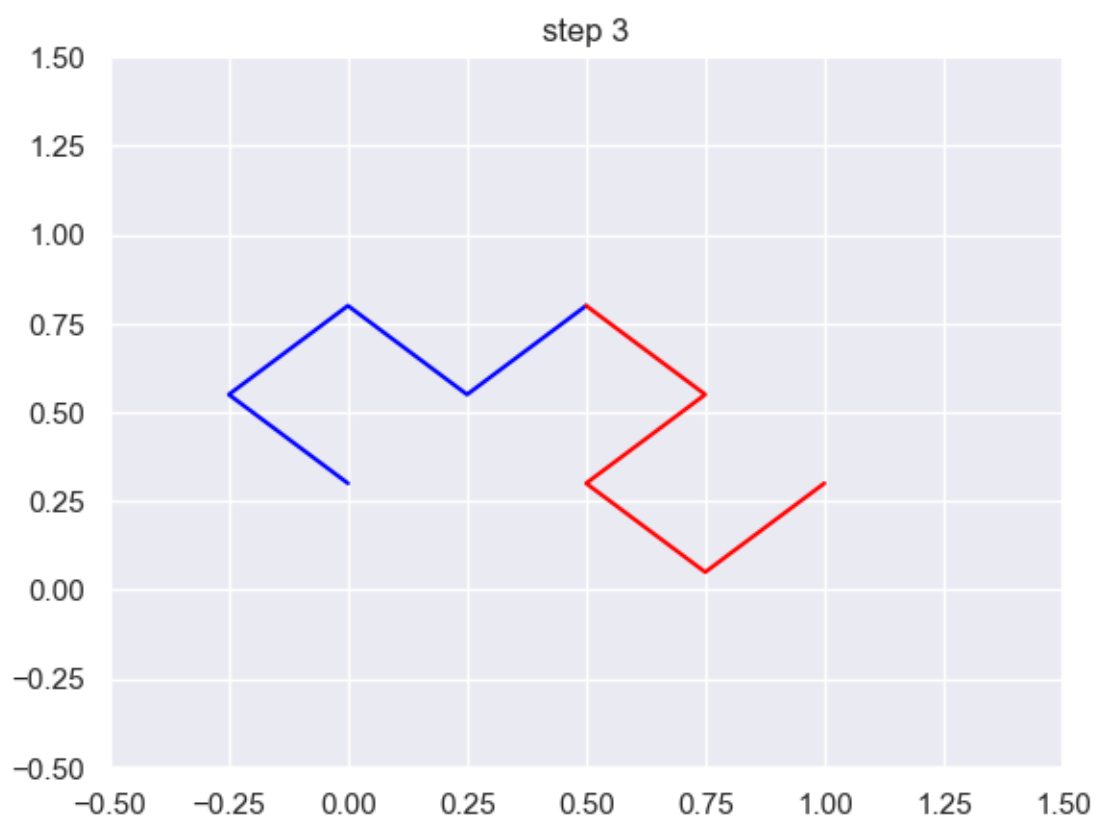
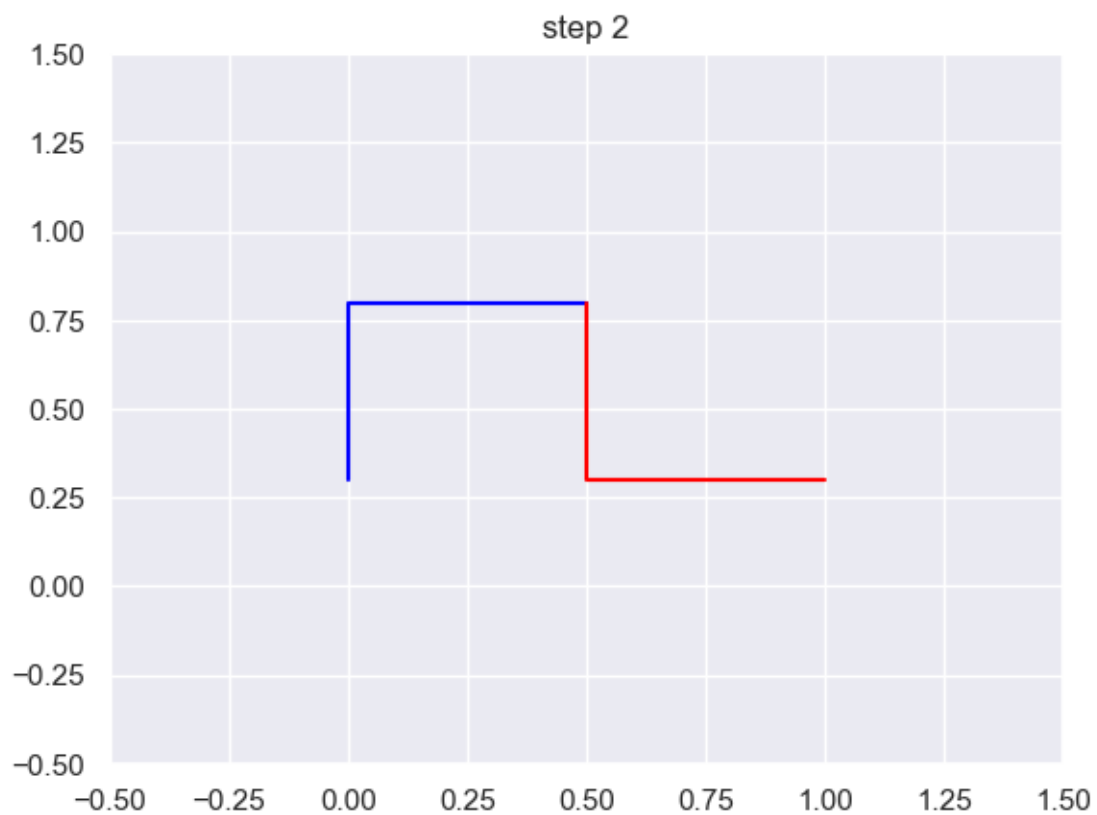
- 1- ابتدا شکل را با ضریب  $\frac{\sqrt{2}}{2}$  کوچک میکنیم (چون هر دو خط روی یک خط ساخته می شوند و مثل اضلاع مربع هستند پس با این ضریب کوچک تر از قطرشان که همان خط قبلی است هستند.) و سپس 45 درجه در خلاف جهت عقربه های ساعت دوران می دهیم.
- 2- دوباره شکل را شکل را با ضریب  $\frac{\sqrt{2}}{2}$  کوچک میکنیم و سپس 135 درجه (90+45) درجه در خلاف جهت عقربه های ساعت دوران می دهیم و سپس با انتقالی انتهایش را به انتهای آخرین خطی که کشیدیم وصل میکنیم.

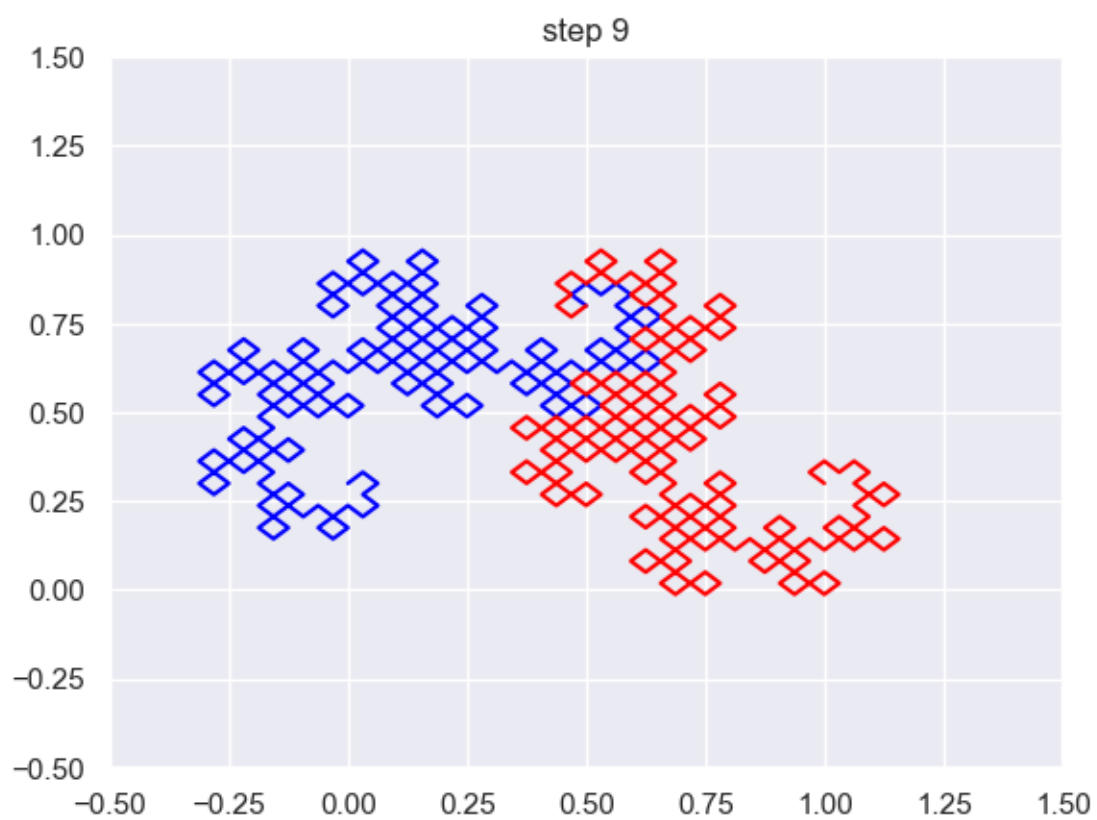
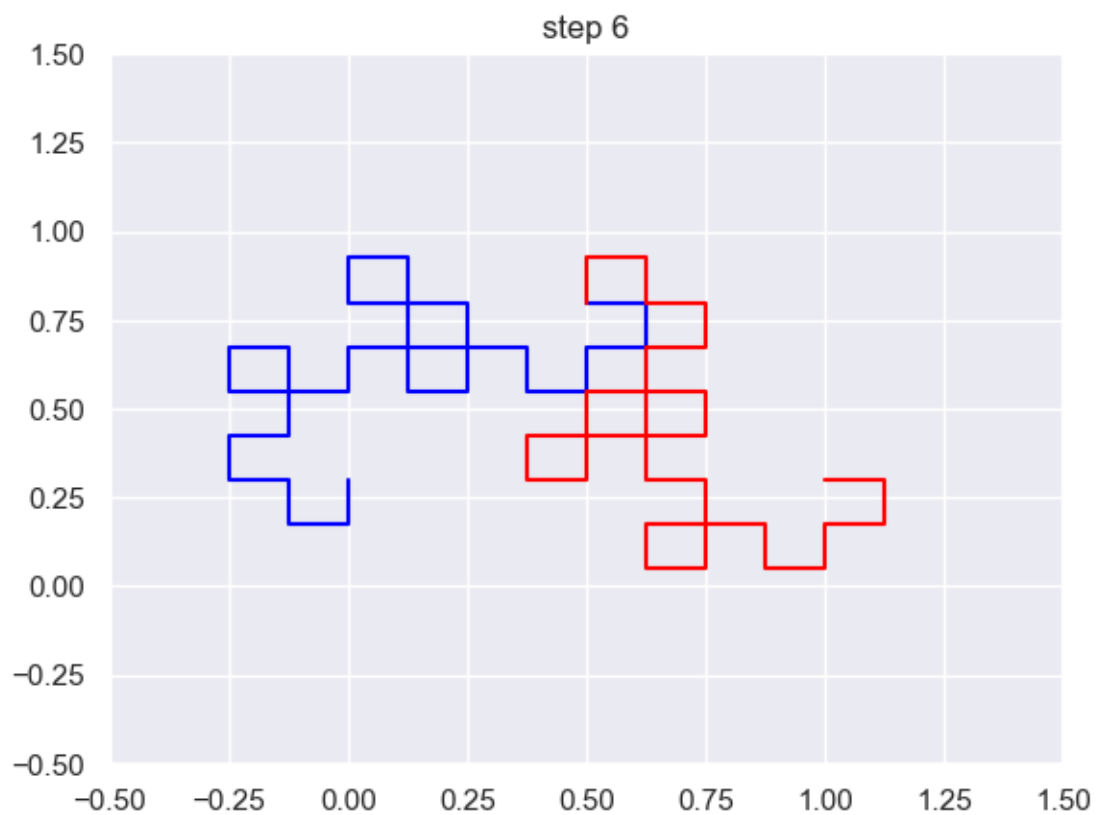
نکات فنی در این جا هم مانند سوال قبل به طور کامل در کد بصورت کامنت گذاری توضیح داده شده است فقط به یک سری نکات مهم اشاره می کنیم:

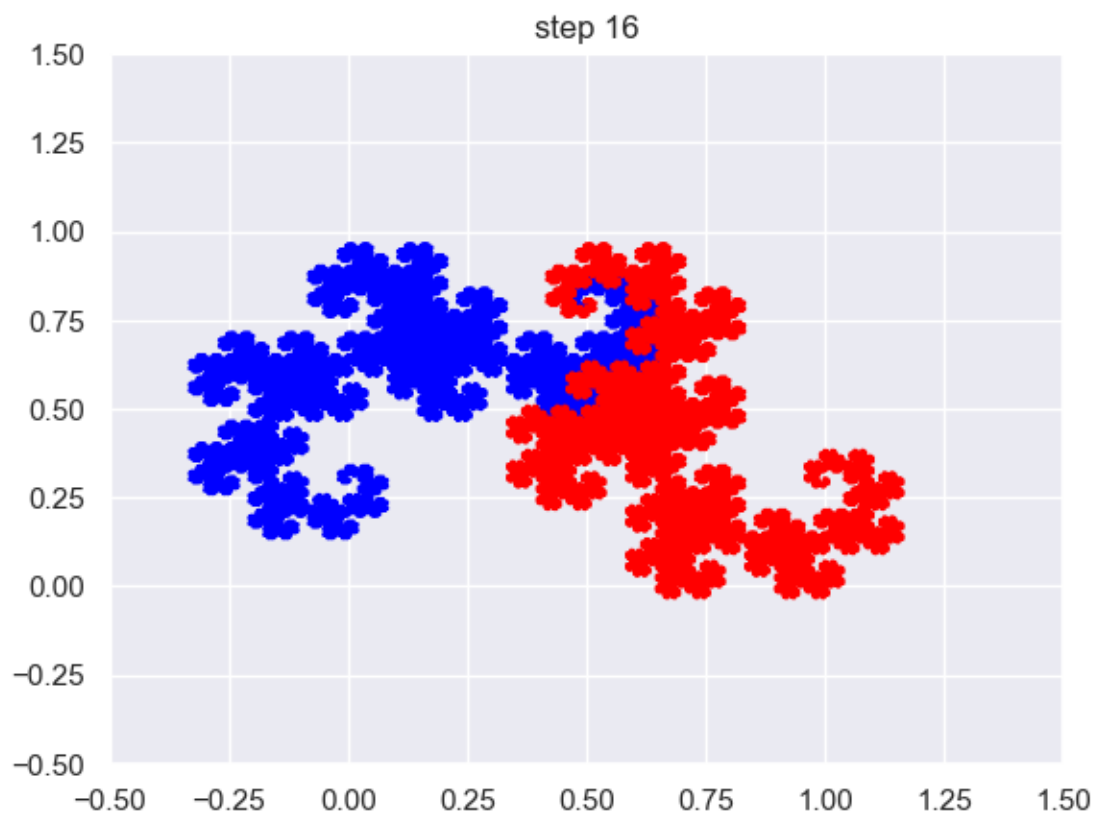
- 1- در این قسمت هم مانند قسمت قبل برای راحتی در نوشتن و دیباگ کردن کد از چند تابع کمکی برای دوران ، انتقال ، اسکیل کردن و ... استفاده کرده ایم.
- 2- در قسمت تابع f2 گفته ایم که انتهای شکل جدید را به انتهای شکل بعدی وصل می کنیم ، پس باید حواسمان به نحوه ی کار کتابخانه ای که برای رسم نمودار استفاده کرده ایم باشد و به این نکته دقت کنیم که این کتابخانه نقاط را به ترتیبی که به آن داده ایم رسم میکند پس برای ذخیره ی مختصات شکل جدید در ادامه ی شکل قبلی باید ابتدا ترتیب مختصات نقاط جدید را عوض کنیم (که این کار را با تابع flip که یک تابع داخلی کتابخانه ی numpy است انجام داده ایم.)
- 3- در این تمرین برای راحتی در همان قسمت تابع f1 که کل شکل را اسکیل کرده ایم مستقیماً نقاط اسکیل کرده را به عنوان نقاط اصلی ذخیره کرده ایم که در تابع بعدی هم مجبور به اسکیل کردن نباشیم.











در این تمرین چون تعداد شکل های نتیجه زیاد بود بعضی هایشان را در این گذارش آورده ایم و بقیه ی شکل ها به طور کامل در فایل مربوط به کد موجود می باشند.

## تمرین 2.3

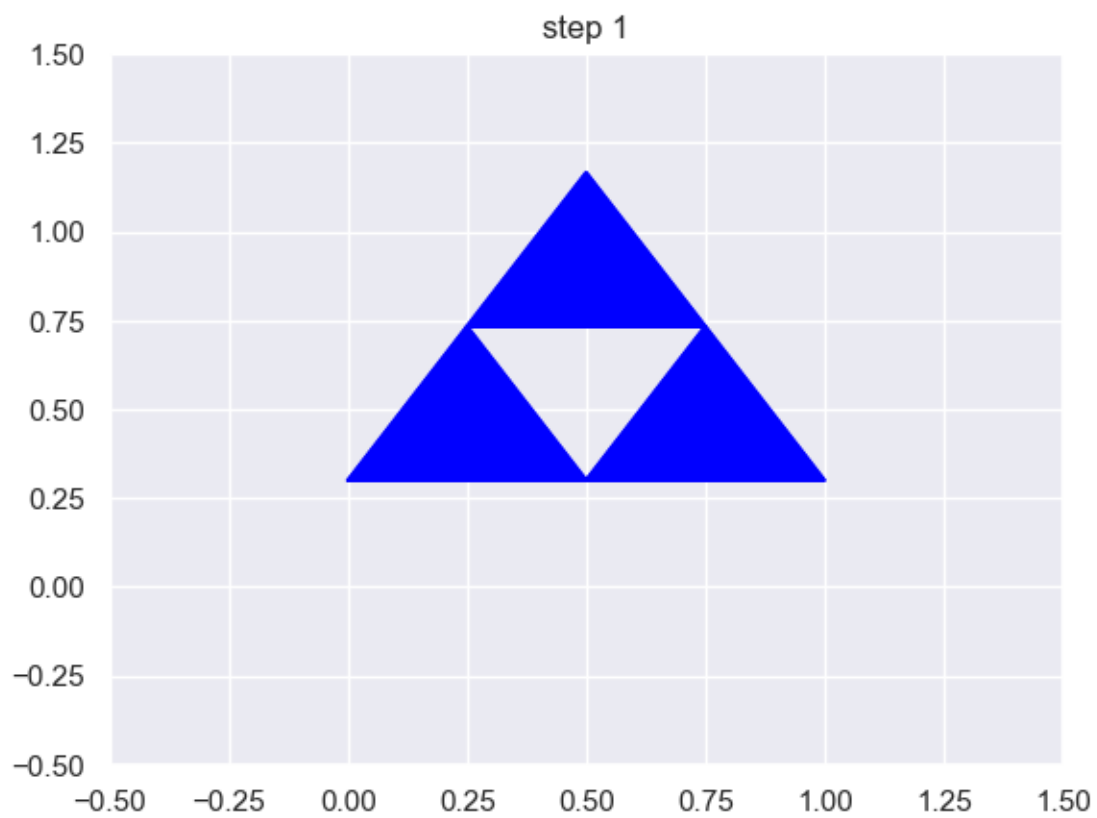
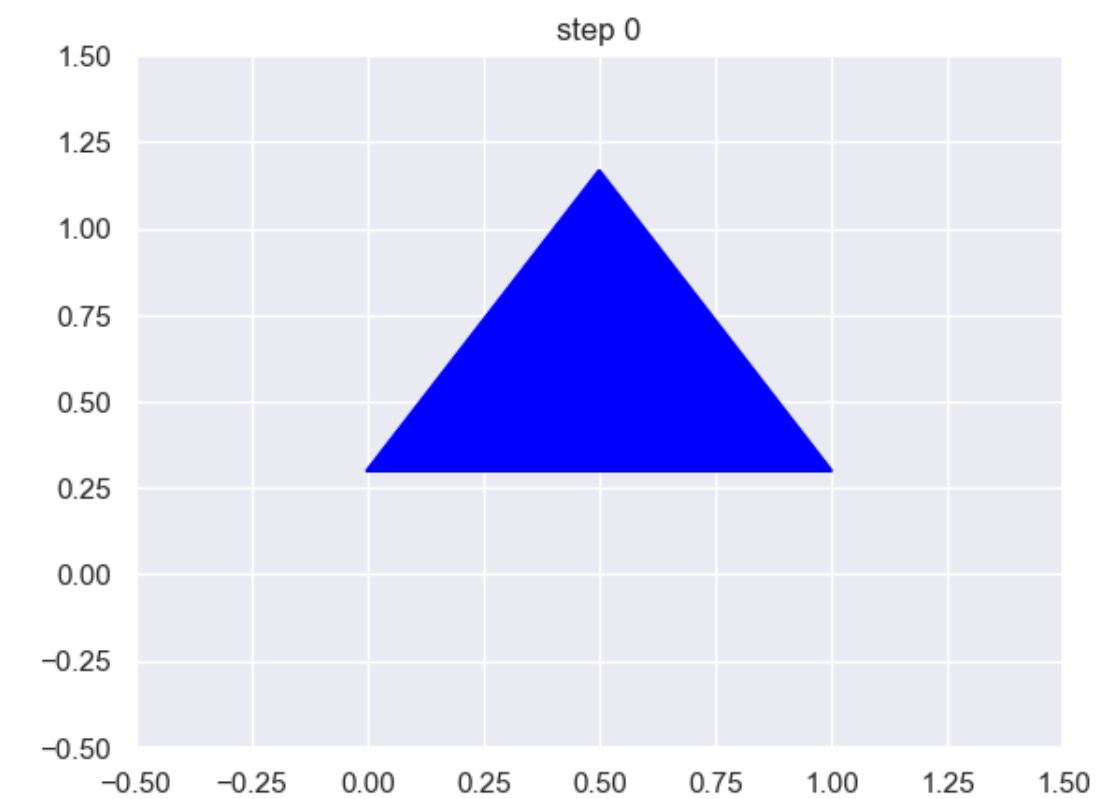
این تمرین هم مانند تمرین های قبل است فقط با این تفاوت ها که:

- 1- در ابتدا بجای یک خط با یک مثلث شروع می کنیم.
- 2- دیگر همه ی نقاط را با هم در یک آرایه نگهداری نمی کنیم چون برای رسمشان به مشکل می خوریم و برای حل این مشکل نقاط مربوط به هر مثلث که باید کشیده شود را در یک آرایه ی زیر مجموعه ی آرایه ی اصلی نگهداری می کنیم که به ما این امکان را می دهد که هر مثلث را مستقل از مثلث های دیگر رسم کنیم.
- 3- تفاوت آخر هم در توابع خود متشابهی است که اعمال می کنیم.

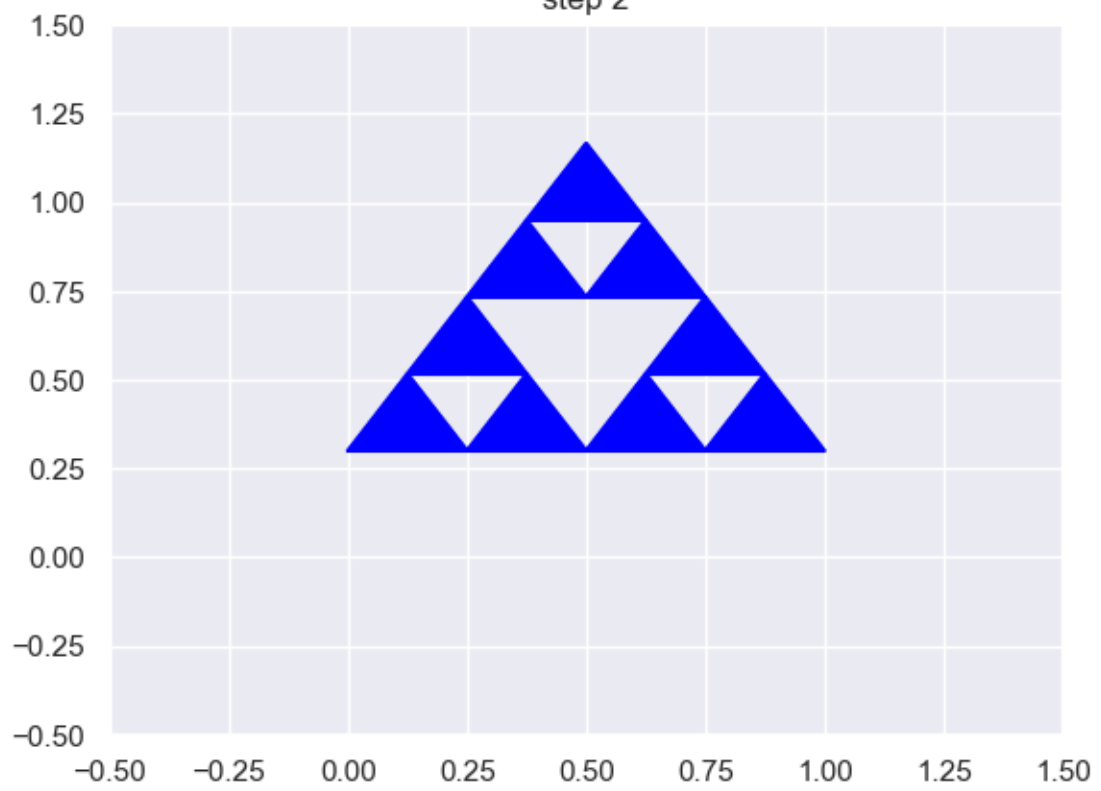
توابع خود متشابه در این قسمت به عبارت زیر اند:

- 1- ابتدا کل نقاط را با ضریب  $\frac{1}{2}$  کوچک می کنیم و داخلش را رنگ می کنیم. (که خود به خود مثلث پایین سمت چپ را تشکیل می دهد).
- 2- همه ی نقاط را با ضریب  $\frac{1}{2}$  کوچک می کنیم و سپس با بردار انتقالی به اندازه ی  $\frac{1}{2}$  طول ضلع مثلث اولیه به سمت راست انتقال می دهیم و داخلش را رنگ می کنیم.
- 3- همه ی نقاط را با ضریب  $\frac{1}{2}$  کوچک می کنیم و سپس با بردار انتقالی به اندازه ی  $\frac{1}{4}$  طول ضلع مثلث اولیه به سمت راست و به اندازه ی  $\frac{1}{2} \sin(60)$  ضربدر طول ضلع مثلث اولیه به بالا انتقال می دهیم و داخلش را رنگ می کنیم.

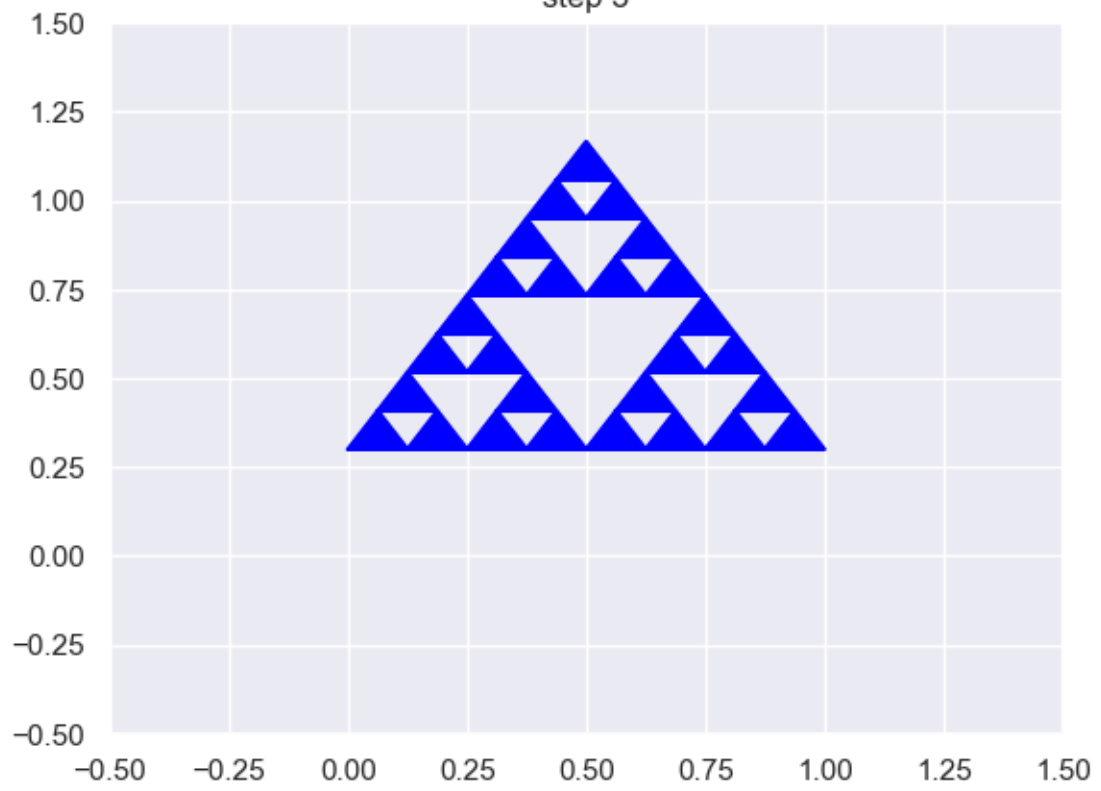
نکات فنی در این جا هم مانند سوال های قبل به طور کامل در کد بصورت کامنت گذاری توضیح داده شده است فقط دوباره باید به یک نکته اشاره کرد که در این قسمت هم مانند قسمت های قبل برای راحتی در نوشتن و دیباگ کردن کد از چند تابع کمکی برای انتقال ، اسکیل کردن و ... استفاده کرده ایم.



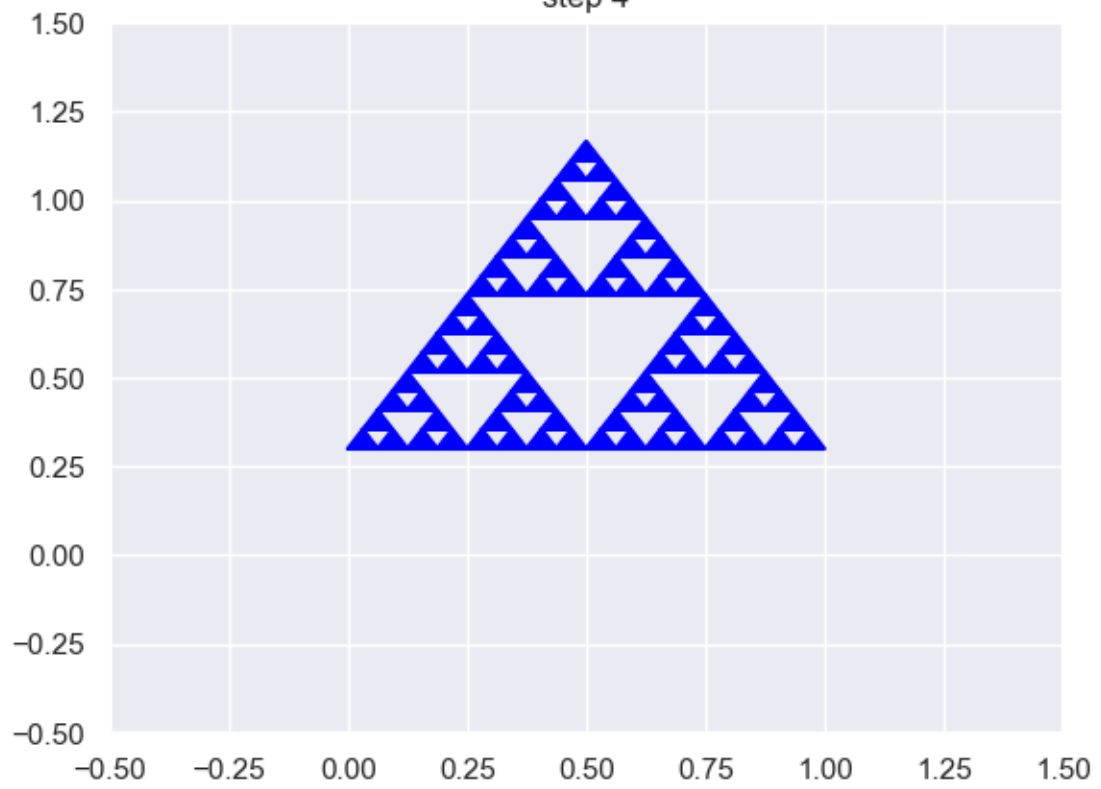
step 2



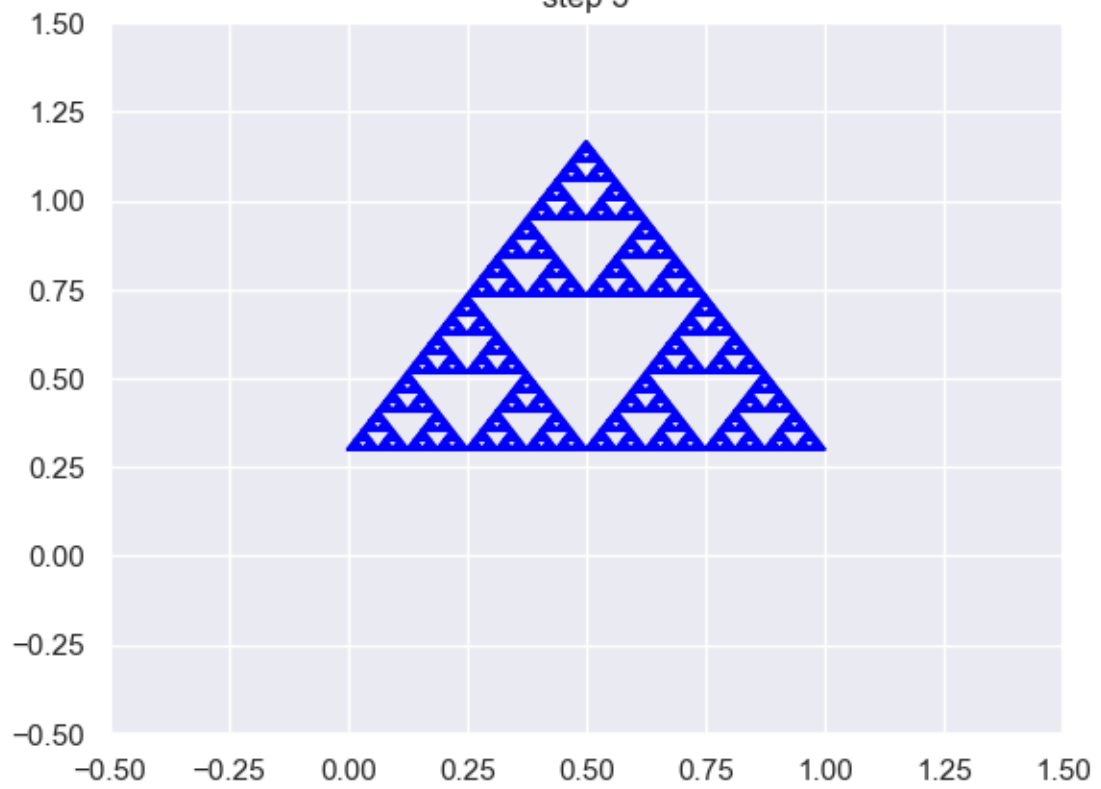
step 3



step 4



step 5



## تمرین 2.4

این تمرین توضیح خاصی ندارد.

در قسمت تابع generate اعداد مربوط به مثلث خیام پاسکال را تولید می کنیم و آن را به صورت یک لیست خروجی می دهیم.

اعداد مربوط به هر مرحله را نیز در قالب یک لیست به عنوان زیرمجموعه ی لیست اصلی اضافه می کنیم. برای تولید اعداد مربوط به هر مرحله ابتدا یک عدد 1 در ابتدای لیست می گذاریم و سپس با یک حلقه ، دوتا دوتا اعداد مرحله قبل را با هم جمع میکنیم و در لیست جدید می ریزیم و سپس با یک 1 در انتهای لیست کار را به اتمام می رسانیم.

دو مرحله ی اول نیز ثابت هستند. ( [1], [1, 1] )

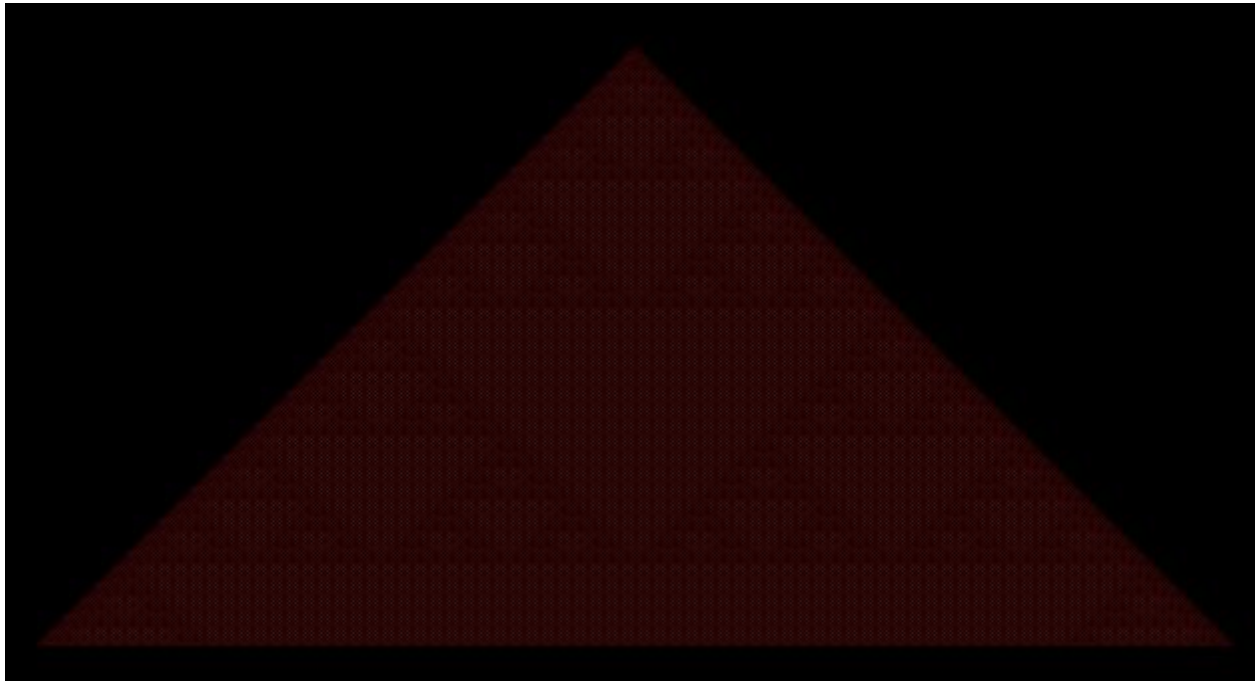
در قسمت تابع draw هم کار خاصی نکردیم فقط از اعداد تولید شده توسط تابع قبلی استفاده کردیم پیکس های مربوط به اعداد زوج (اعدادی که باقی مانده تقسیمشان بر عدد 2 ، صفر می شود) را با رنگ قرمز و پیکسل های مربوط به اعداد فرد را با رنگ سبز نمایش داده ایم.

برای نمایش دقیق هر پیکسل در جای مخصوص خودش هم در هر مرحله به ازای هر عضو موجود در آن ابتدا یکی را می کشیدیم سپس یک پیکسل فاصله گذاشته و بعدی را می کشیدیم ( در راستای محور x ها حرکت می کردیم) و در پایان هر مرحله هم یک واحد به y ها اضافه می کردیم و دوباره شروع به کشیدن پیکسل ها می کردیم.

از متغیر last\_step\_starting\_x هم برای ثبت نقطه ی شروع x های مرحله های قبل استفاده می کردیم.

در نهایت نتیجه مانند مثل سرپینسکی شد.





شکل زیر هم با کتابخانه ی matplotlib ایجاد شده است که بجای پیکسل از دایره استفاده کرده ایم ولی چون خیلی اسکیل دایره ها کوچک شد مثل پیکسل شده است.

