

# گزارشکار آزمایشگاه شبیه سازی رایانه ای

مهران آبالان

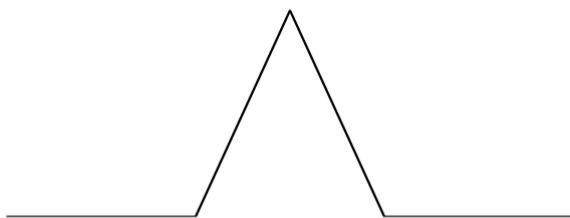
403201223

برفدا نه کخ: میتوان با چهار گام این فرآکتال را ساخت:

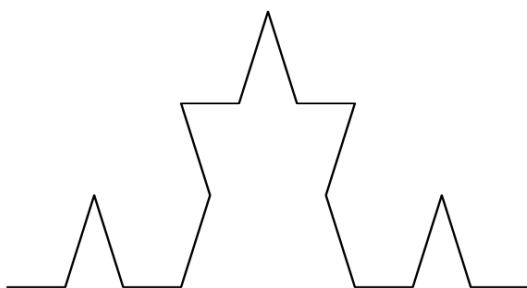
1. شکل اولیه را با مقیاس یک سوم اسکیل میکنیم.
2. یک کپی از شکل اسکیل شده را به اندازه 60 درجه میچرخانیم و با یک انتقال به انتهای سمت راست شکل اسکیل شده مبیریم.
3. این بار شکل اسکیل شده را به اندازه منفی 60 درجه میچرخانیم و دوباره این شکل را به اندازه ای انقال میدهیم که از انتهای شکل گام قبل شروع شود.
4. یک کپی از شکل گام اول را به انتهای شکل گام سوم منتقل میکنیم.

همه این گام ها را با دو تابع میتوان انجام داد، یک تابع برای دوران یک بردار به اندازه دلخواه و یک تابع برای انتقال به اندازه یک بردار دلخواه. با کنار هم گذاشتن این توابع ساخته شده و گام های بالا میتوان برفدا نه را تولید کرد:

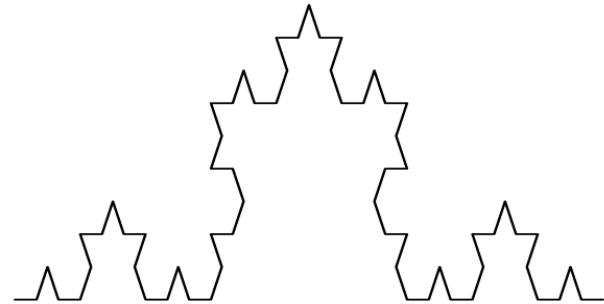
شکل اولیه



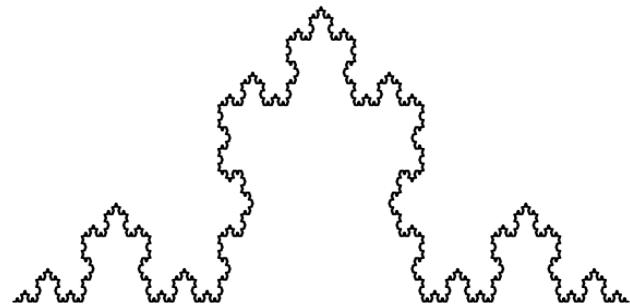
یک بار اعمال 4 گام بالا



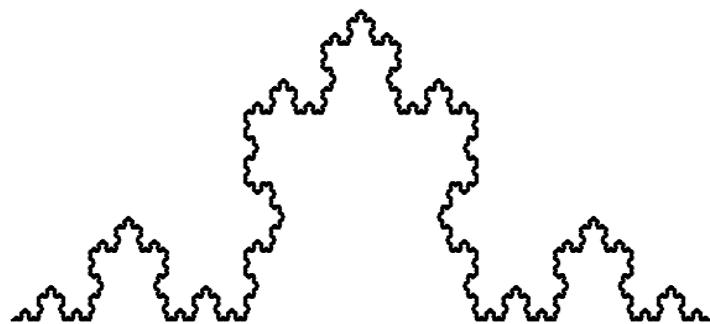
دو بار اعمال 4 گام بالا



3 بار اعمال 4 گام بالا



5 بار اعمال 4 گام بالا

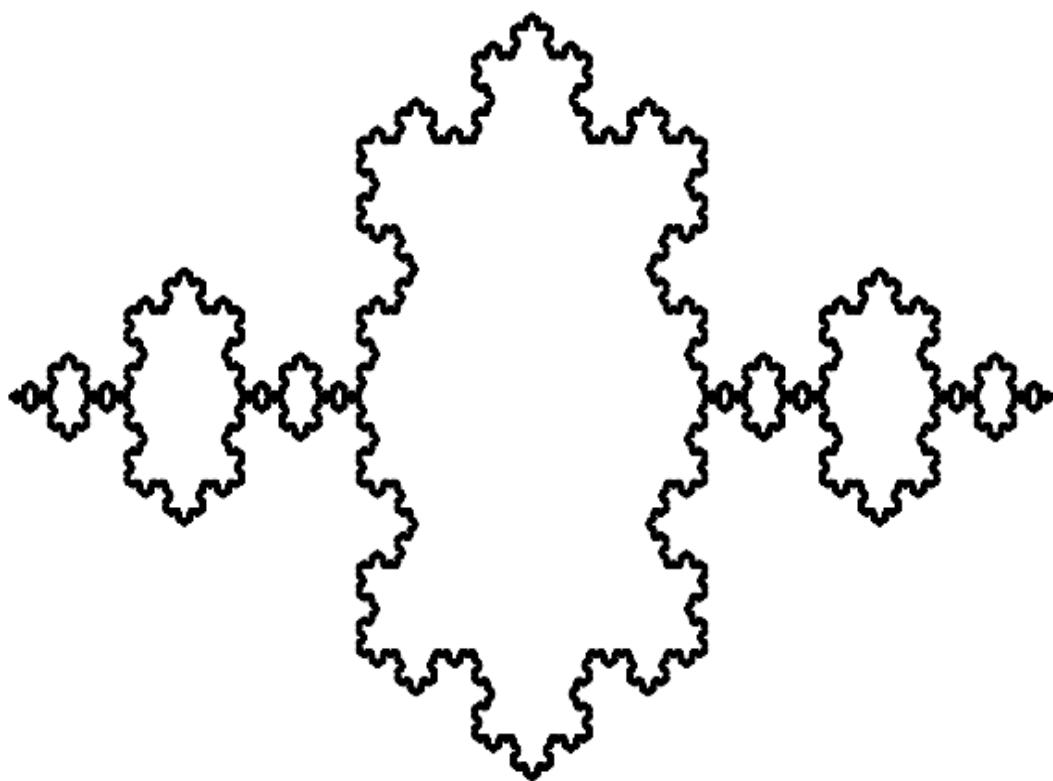


8 بار اعمال 4 گام بالا

در کد حل سوال، خروجی برای چند حالت جلوتر هم نشان داده شده است ولی اساسا فرق زیادی با این شکل نخواهد داشت.

برای تولید کل برفدانه بازتاب شکل بالا نسبت به محور افقی به آن اضافه کردم. میتوانستیم یک مثلث هم به ورودی پاس بدم تا نوع دیگر برفدانه حاصل شود.

Koch Snowflake-order 10



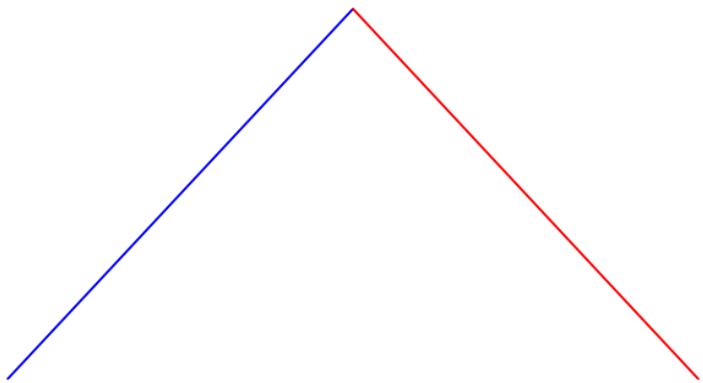
اژدهای هی وی: برای ساخت این فراکتل دو گام زیر کافی است:

1. شکل را به اندازه یک بر روی رادیکال دو اسکیل می کنیم و سپس 45 درجه می چرخانیم.
2. شکل را دوباره به اندازه یک بر روی رادیکال دو اسکیل می کنیم سپس حول محور افقی بازتاب آن را حساب می کنیم و به اندازه منفی 45 درجه آنرا می چرخانیم.

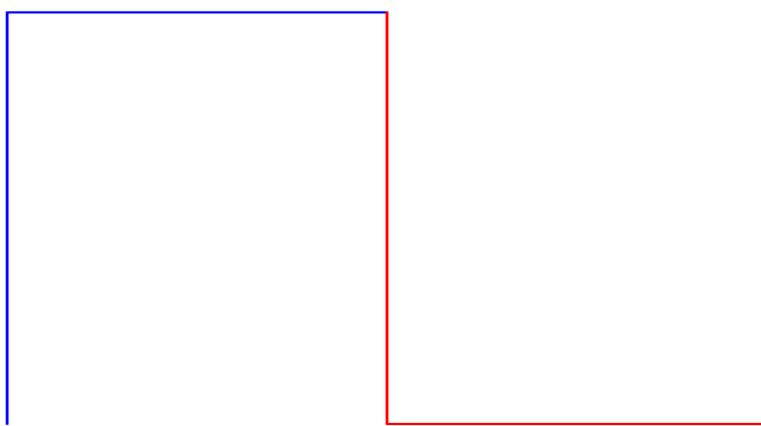
اصولاً کد من هم گام بالا را انجام میدهد فقط کمی دیدگاه متفاوتی را نشان میدهد، در داخل کد دو نقطه متوازی را گرفته ام و بردار فاصله این دو را بدست آوردم و این بردار را بسته به قرد یا زوج بودن یک شمارنده یا به اندازه مثبت 45

یا منفی 45 دوران داده ام و در نهایت این بردار دوران یافته را اسکیل کردم و یک نقطه جدید برسست آوردم که بین دو نقطه قبلی(دو نقطه ای که این نقطه رو باهش ساختم) قرار دادم و در نهایت با اعمال متوالی این کار اژدها ساخته شد!

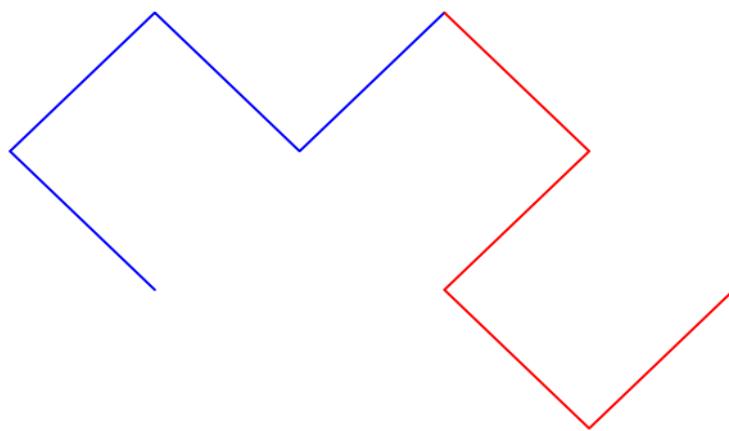
Heighway Dragon Fractal - Order 1



Heighway Dragon Fractal - Order 2

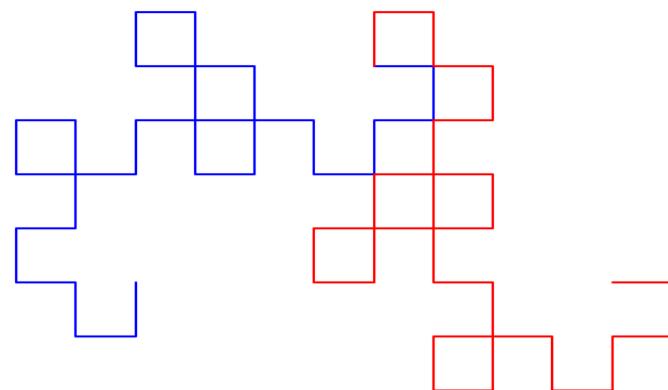


Heighway Dragon Fractal - Order 3



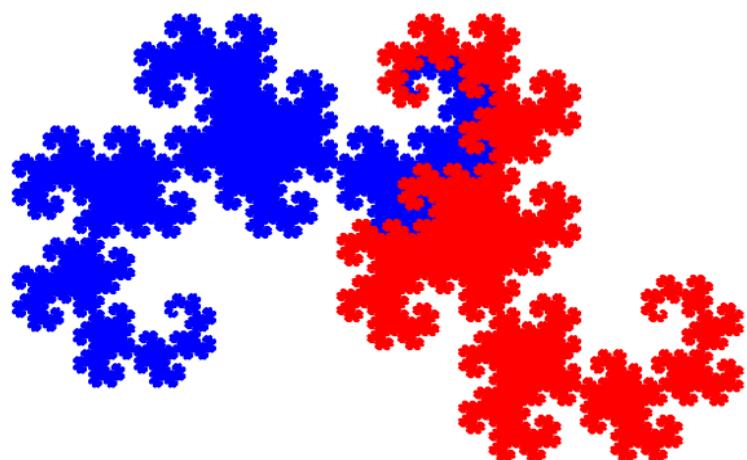
پس از شش بار تکرار گام ها به شکل زیر میرسیم:

Heighway Dragon Fractal - Order 6



پس از هفده بار تکرار ازدها ساخته میشود:

Heighway Dragon Fractal - Order 17



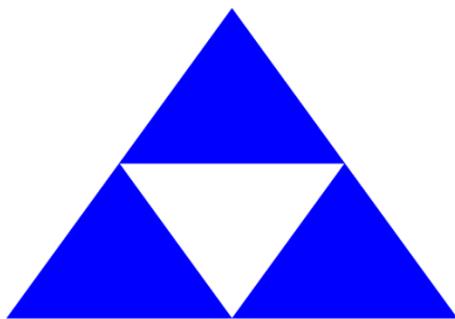
در خود کد خروجی برای مرتبه های دیگر هم نمایش داده شده است.

**مثلث سرپینسکی:** برای ساخت این مثلث آمده ام و سه گام زیر را در کد برداشته ام و مثلث را ساختم.

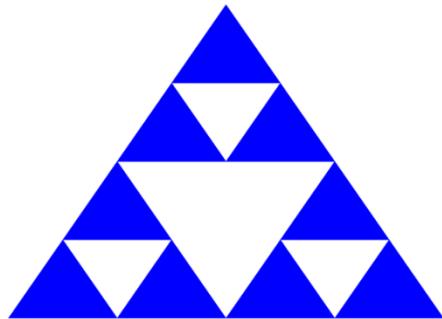
1. اسکیل شکل اولیه به اندازه 0.5.
2. انتقال یک کپی از این شکل اسکیل شده به راس بالایی شکل گام 1.
3. انتقال یک کپی از شکل گام 1 به راس راست مثلث اسکیل شده گام 1.

در نهایت شکل های حاصل به این صورت اند:

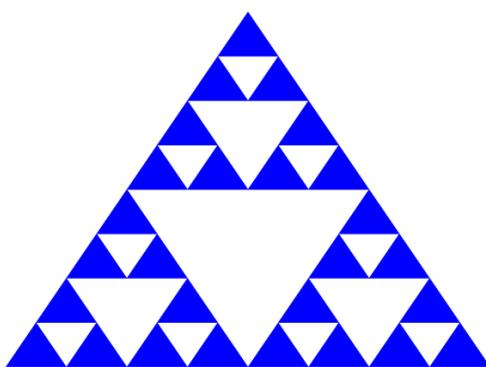
Sierpinski Fractal - Order 1



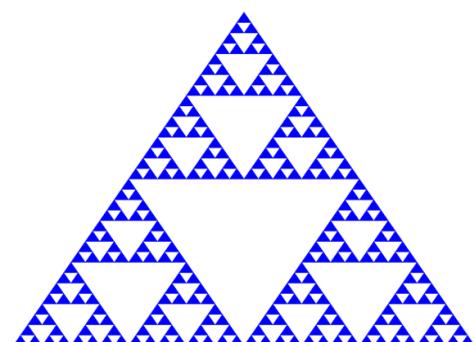
Sierpinski Fractal - Order 2



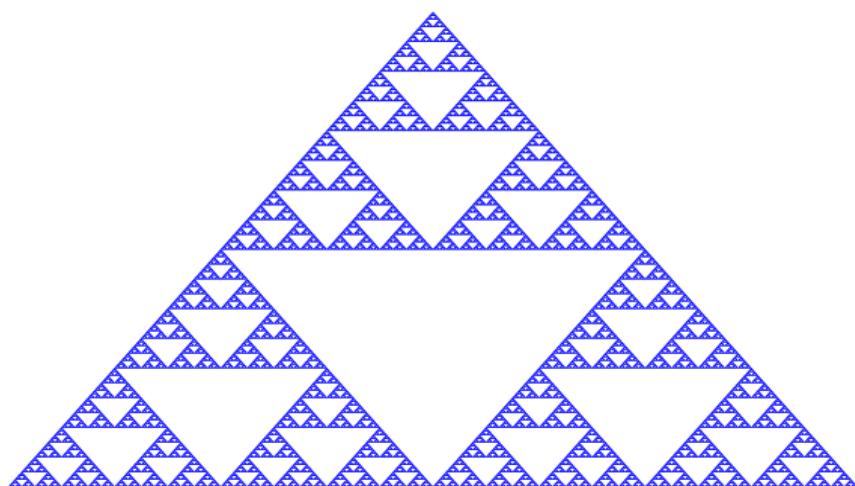
Sierpinski Fractal - Order 3



Sierpinski Fractal - Order 5



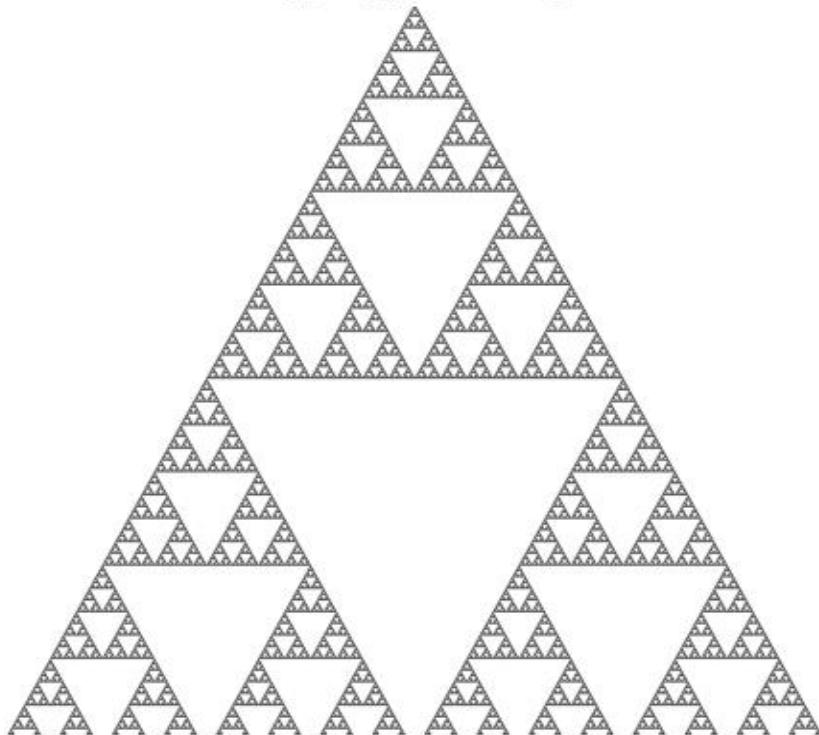
Sierpinski Fractal - Order 8



**مثلث خیام-پاسکال:** در این سوال هدف تولید مثلث خیام پاسکال و هم رنگ کردن عناصر زوج و فرد بود تا نشان دهیم خروجی این کد هم مثلث سرپینسکی را می‌سازد.

برای تولید مثلث خیام پاسکال تا گام  $n$  ام یک آرایه  $n * n$  ساختم با عناصر صفر و عناصر زیر قطر اصلی رو از بسط دو جمله ای ساختم و در نهایت در سطر عناصر ناصفر را به وسط سطر منتقل کردم. چون فقط اعداد زوج و فرد را میخواستیم با قی مانده همه اعداد آرایه به 2 را نگه داشتم و در نهایت این آرایه را کشیدم.

khayyam\_pascal triangle



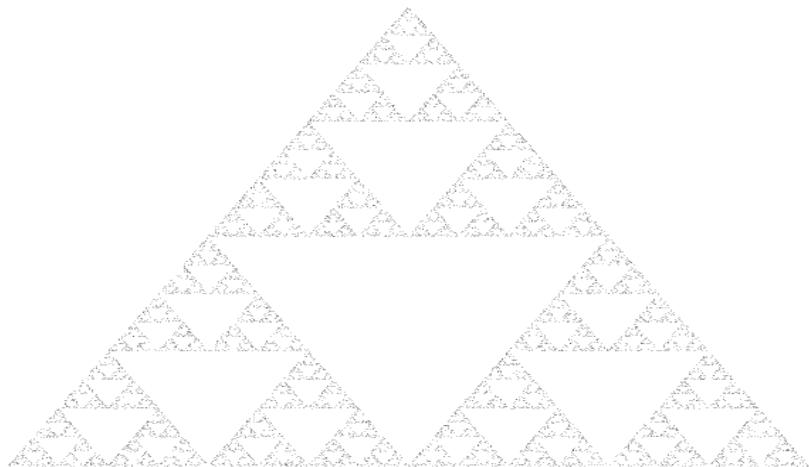
ران تایم این الگوریتم از الگوریتم قبلی برای تولید مثلث سرپینسکی بسیار سریع تر بود و طهمان طور که در شکل میبینیم به مثلث سرپینسکی تبدیل شده است.

**باز هم مثلث سرپینسکی:** در این سوال سه گامی که برای مثلث سرپینسکی برداشتیم را با احتمال مساوی روی نقاط رندم در صفحه انجام دادیم و پس از  $p$  بار اعمال هر کدام از آن توابع و تعداد نقاط کافی مثلث سرپینسکی پدید آمد. خروجی کد برای 1000 نقطه، 10000 نقطه و یک میلیون نقطه را نمایش داده ام، در داخل کد خروجی برای نقاط دیگر هم میتوانید مشاهده کنید. برای کمتر شدن ران تایم از محاسبات برداری آرایه های نامپای استفاده کردم و یک لوپ را حذف کردم و درنتیجه ران تایم کد بسیار پایین تر آمد.

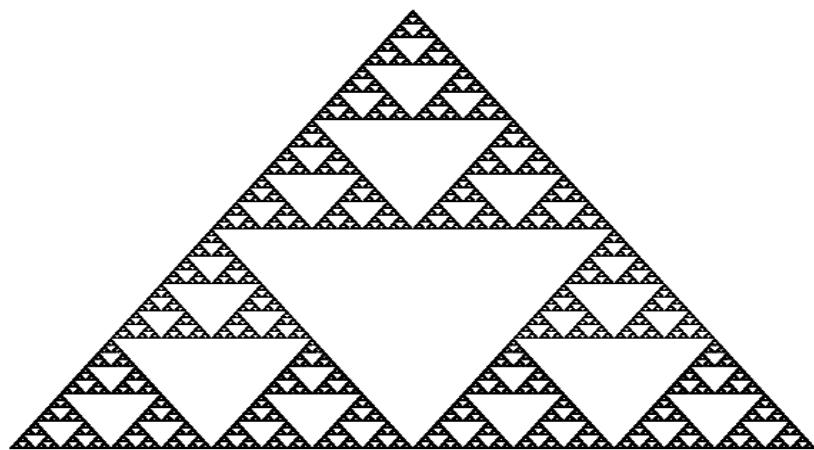
Sierpinski triangle with 1000 points



Sierpinski triangle with 10000 points

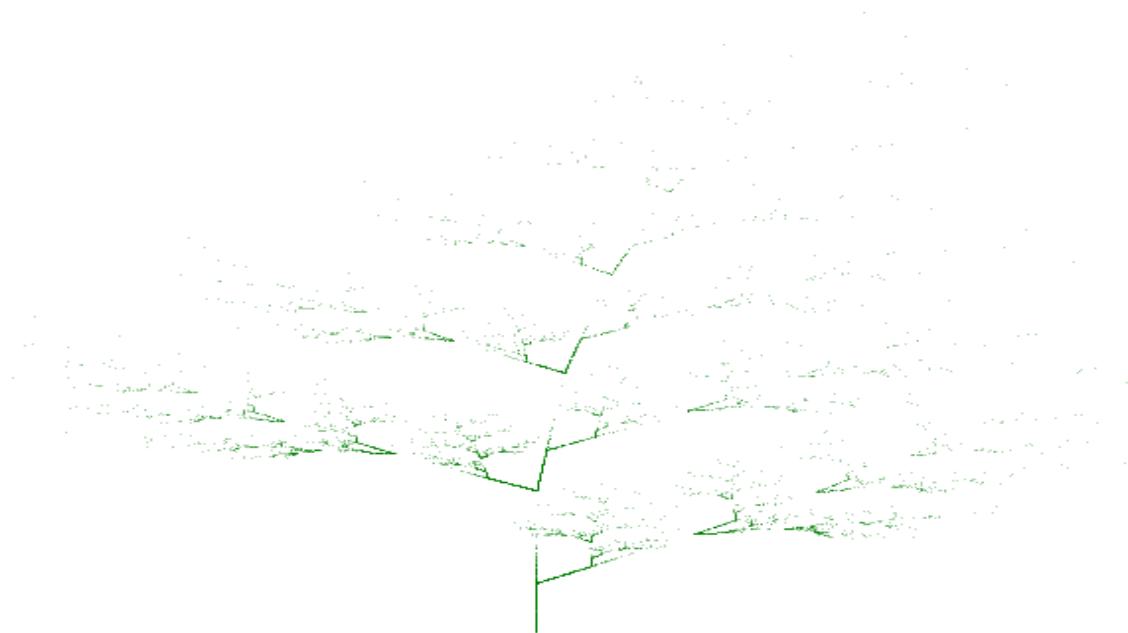


Sierpinski triangle with 1000000 points

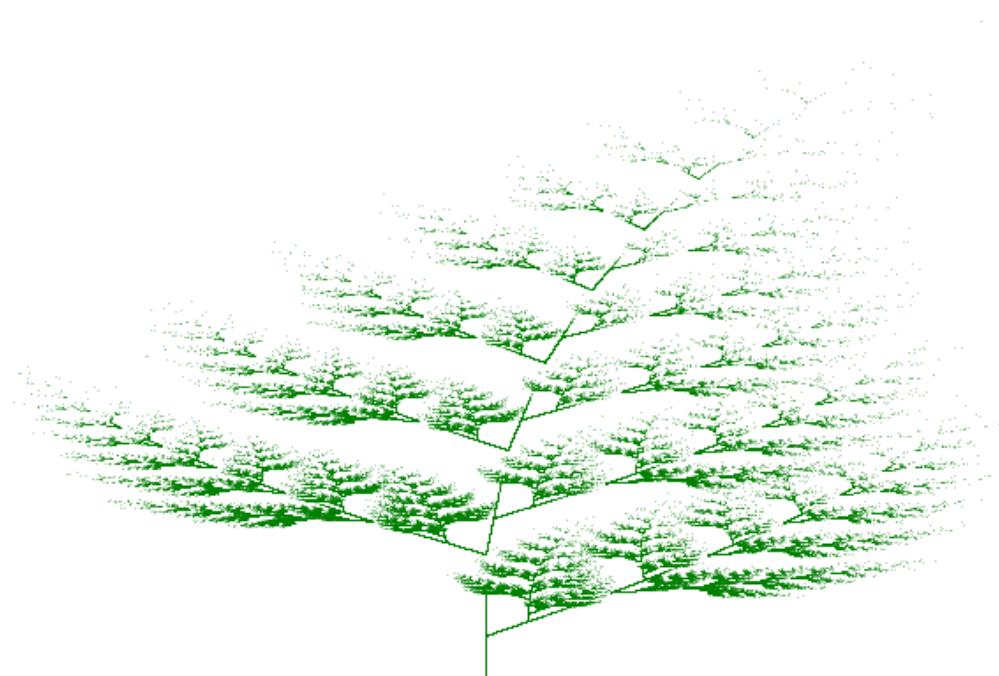


**برگ سرخس:** برای تولید برگ سرخس از چهارتابع مولد که در کامنت های کد با جزئیات کامل توضیح دادم استفاده کردم. مانند بالا این توابع را روی نقاط رندم ولی با توزیع احتمال برابر اثر دادم و خروجی شکل به صورت زیر شد:

Fern\_leaf with 10000 points



Fern\_leaf with 1000000 points



سپس پس از مطالعه در مورد فراکتال برگ سرخس از توزیع احتمال مناسب استفاده کردم و نتیجه درست بدست آمد:

Fern\_leaf with 100000 points

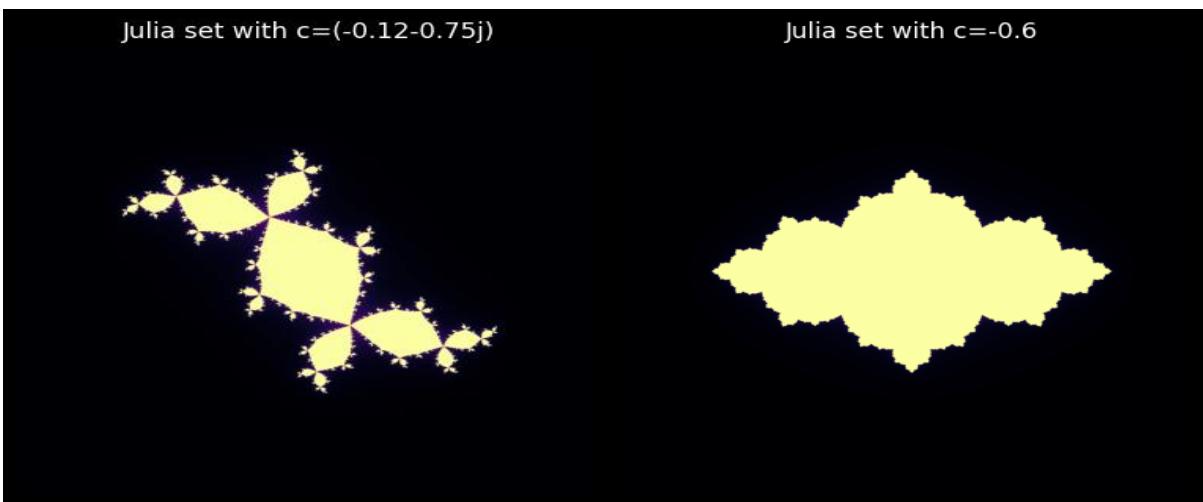
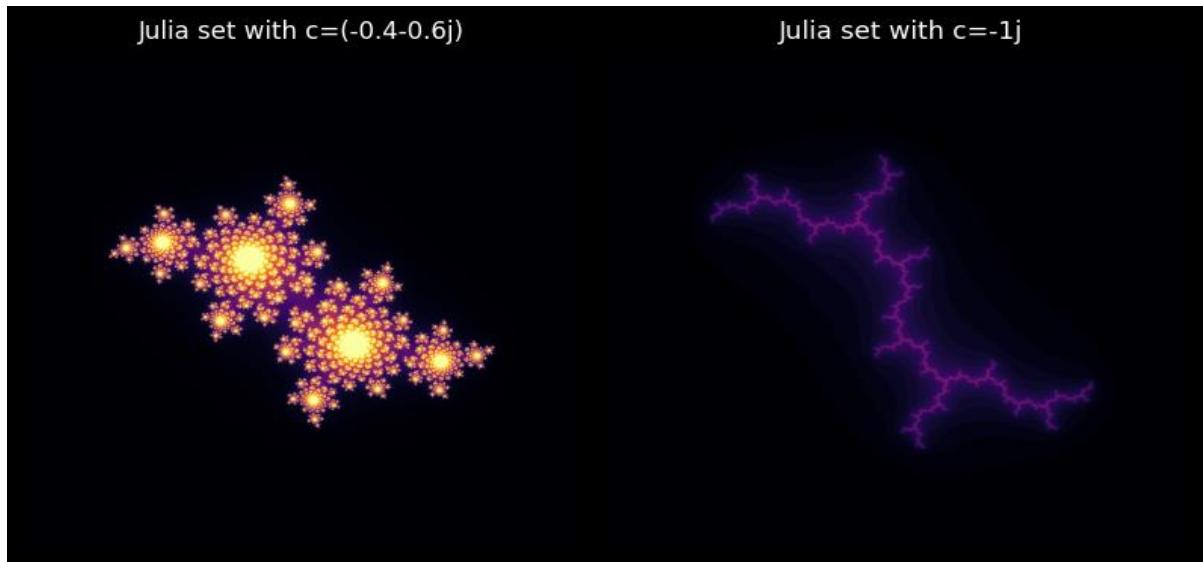


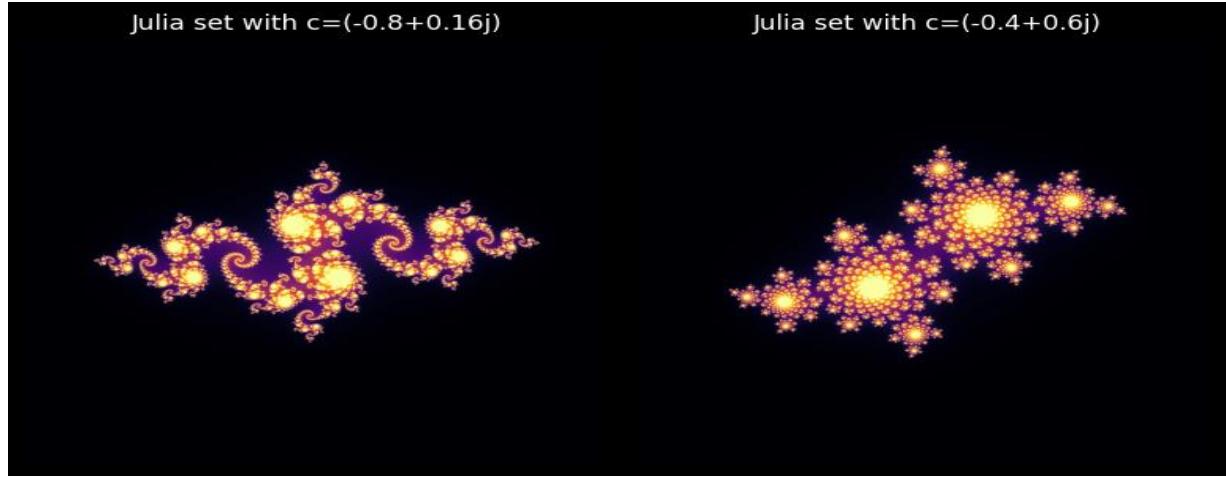
Fern\_leaf with 1000000 points



**مجموعه ژولیا:** تولید این مجموعه به نسبت ساده بود و تابع تعریف شده در درسنامه را هر بار با ثابت دلخواه روی نقاطی که به بی نهایت میل نکردند اثر دادم، برای سریع تر شدن سرعت محاسبه فقط روی نقاطی که داخل محدوده بودند تابع را اعمال کردم.

در ابتدا میخواستم مجموعه را با نقاط انتخابی کاملاً رندم از صفحه بسازم که این مشکل ایجاد میکرد چون توزیع ها نا یکنواخت میشد و اگر تعداد را به اندازه کافی بالا میردم تا توزیع بهتر شود و این نا یکنواختی کمتر شود، ران تایم کد بشدت بالا میرفت، برای همین روی توزیع نقاطی که خودم مشخص کردم تابع را اثر دادم و شکل های خروجی بدست آمد:





داخل کد برای چند ثابت دیگر هم اثر اعمال تابع را بررسی کرد و میتوانید آنجا کل خروجی ها را ببینید. به طور کلی سعی کردم سوال هایی که کامنت کمتری داخل کد گذاشته ام را بیشتر توضیح دهم و این توضیحات به نوعی تکمیل کننده کامنت ها حین انجام هر سوال هست. برای درک بهتر همزمان کد ها هم بخوانید.