

به نام داناترین

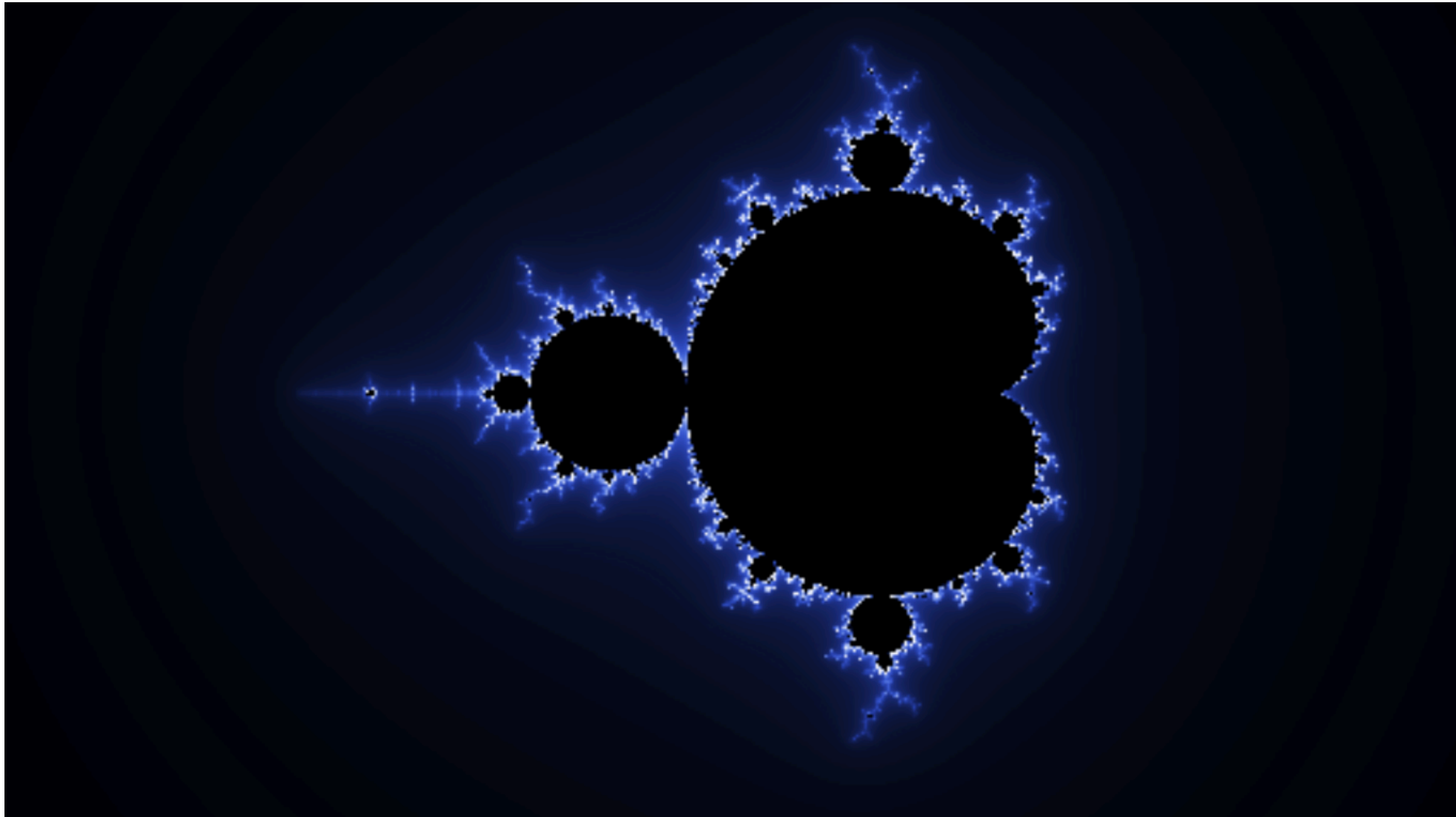


دوره‌ی خلاقیت الگوریتمی و برنامه‌نویسی پایتون

نقاشی‌های فرکتالی

دانشکده‌ی مهندسی کامپیوتر دانشگاه صنعتی شریف

تابستان ۱۴۰۲



fractal-visualizer.netlify.app

zazow.com

• فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.



- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱							

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱						

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲					

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲					

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳				

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳				

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵			

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵			

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵	۸		

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵	۸		

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵	۸	۱۳	

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵	۸	۱۳	

- فرض کنید روز ۱ در حساب بانکی خود ۱ تومان دارید.
- اگر موجودی حساب شما در روز n برابر F_n تومان باشد، بانک دو روز بعد همین مبلغ را به عنوان سود به حساب شما اضافه می‌کند.
- موجودی حساب شما در روز n چقدر است؟

روز	۱	۲	۳	۴	۵	۶	۷	۸
موجودی	۱	۱	۲	۳	۵	۸	۱۳	۲۱

می‌توانید یک رابطه ارائه دهید؟

$$F_n =$$

می‌توانید یک رابطه ارائه دهید؟

$$F_n = F_{n-1} + F_{n-2}$$

$$F_n = F_{n-1} + F_{n-2}$$

```
def F(n):
```

$$F_n = F_{n-1} + F_{n-2}$$

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

```
x = F(5)
```

$$F_n = F_{n-1} + F_{n-2}$$

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

```
F(3) => return F(2)+F(1)
```

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

```
F(3) => return F(2)+F(1)
```

```
F(2) => return F(1)+F(0)
```

فرض کنید $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

```
F(3) => return F(2)+F(1)
```

```
F(2) => return F(1)+F(0)
```

```
F(1) => return F(0)+F(-1)
```

```
def F(n):
```

```
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

```
def F(n):  
    if n <= 2:  
  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$


```
def F(n):  
    if n <= 2:  
        return 1  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

دوباره $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    if n <= 2:  
        return 1  
    return F(n-1)+F(n-2)
```

$$F_n = F_{n-1} + F_{n-2}$$

دوباره $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    if n <= 2:  
        return 1  
    return F(n-1)+F(n-2)
```

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

```
F(3) => return F(2)+F(1)
```

$$F_n = F_{n-1} + F_{n-2}$$

دوباره $F(5)$ را فراخوانی کنیم.

```
def F(n):  
    if n <= 2:  
        return 1  
    return F(n-1)+F(n-2)
```

```
x = F(5)
```

```
F(5) => return F(4)+F(3)
```

```
F(4) => return F(3)+F(2)
```

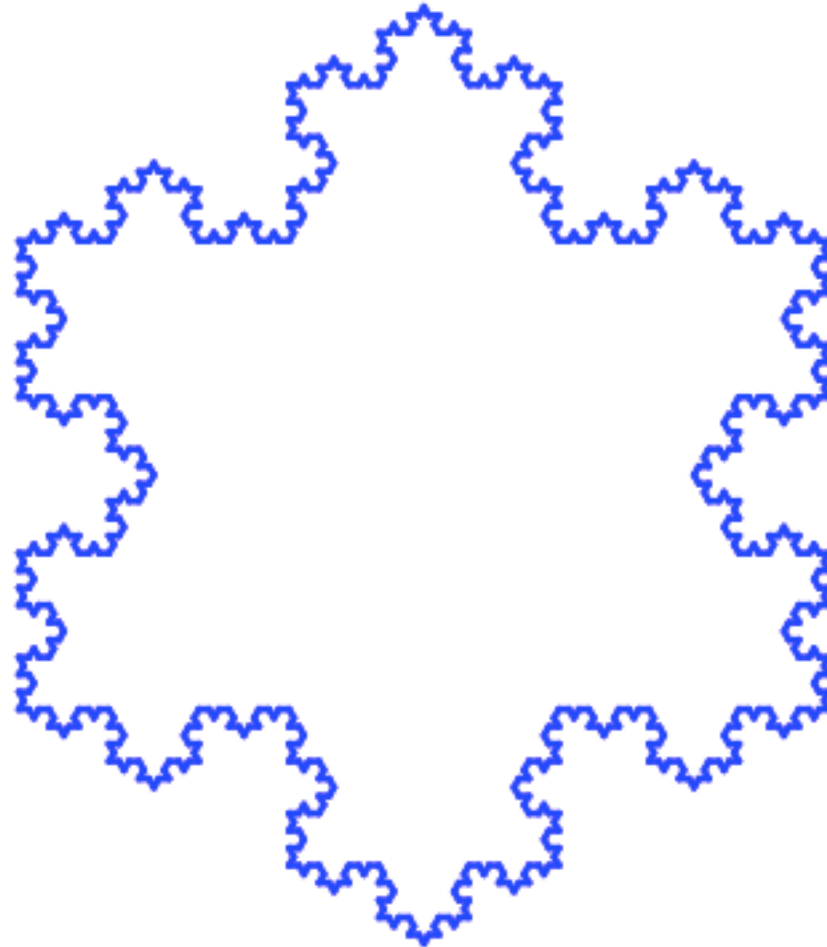
```
F(3) => return F(2)+F(1)
```

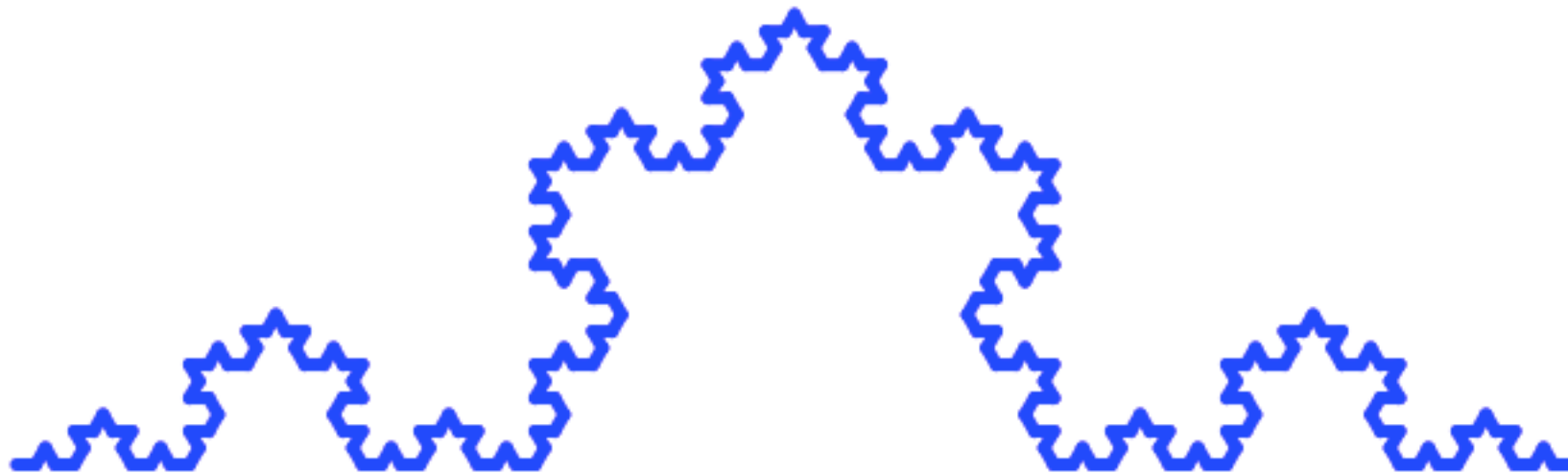
```
F(2) => return 1
```

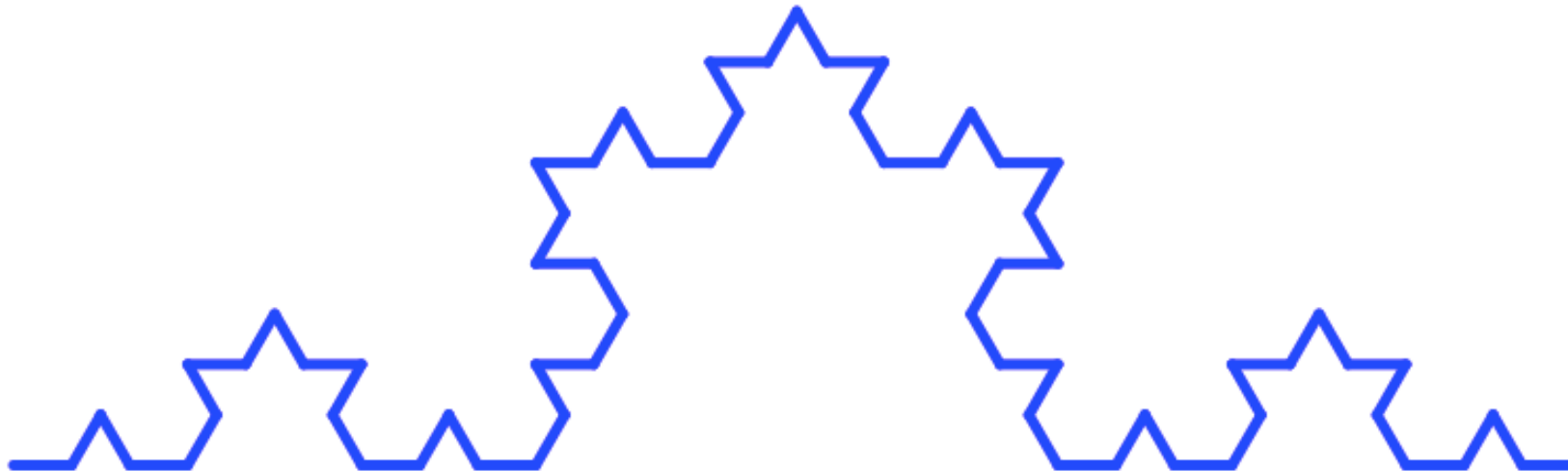
```
F(1) => return 1
```

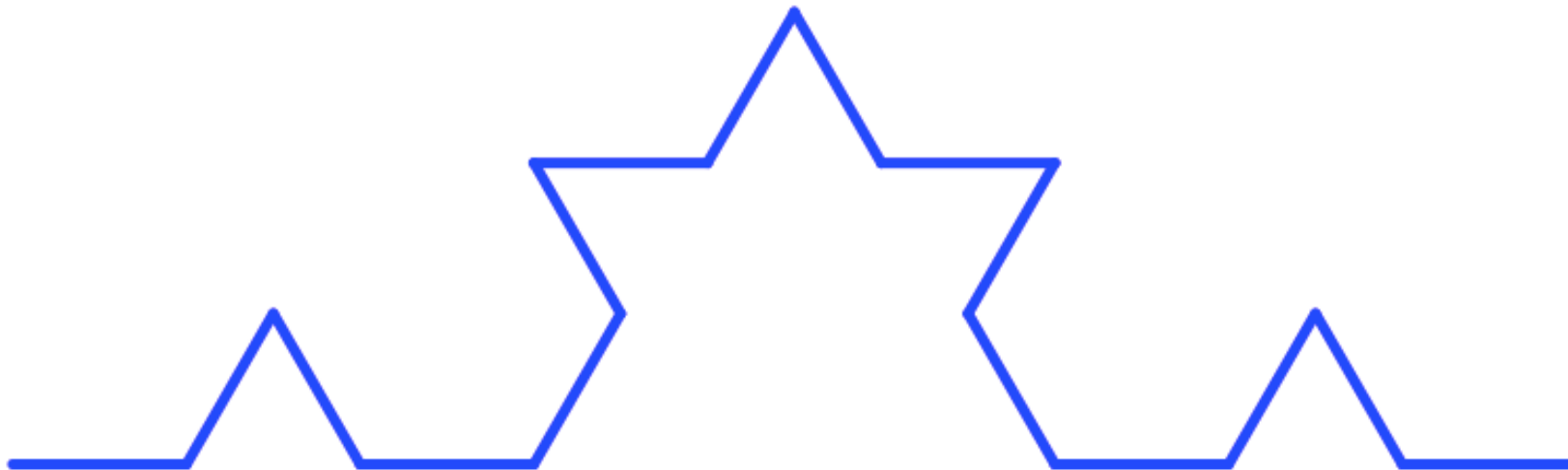
$$F_n = F_{n-1} + F_{n-2}$$

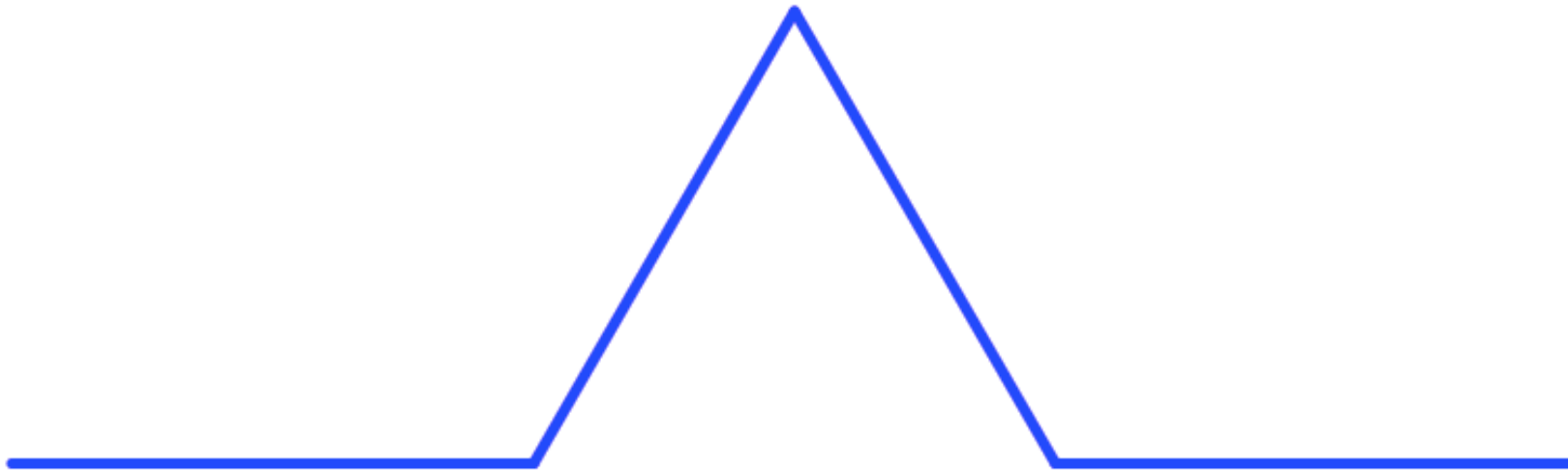
وقت کدزدن (:







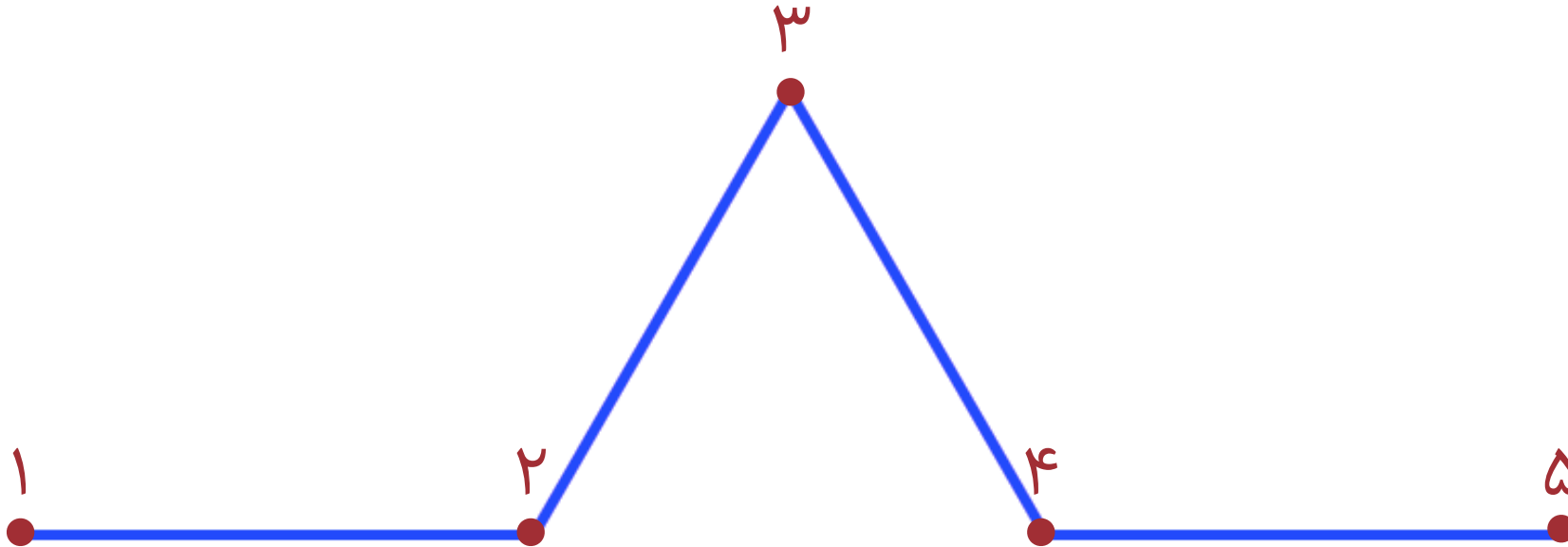




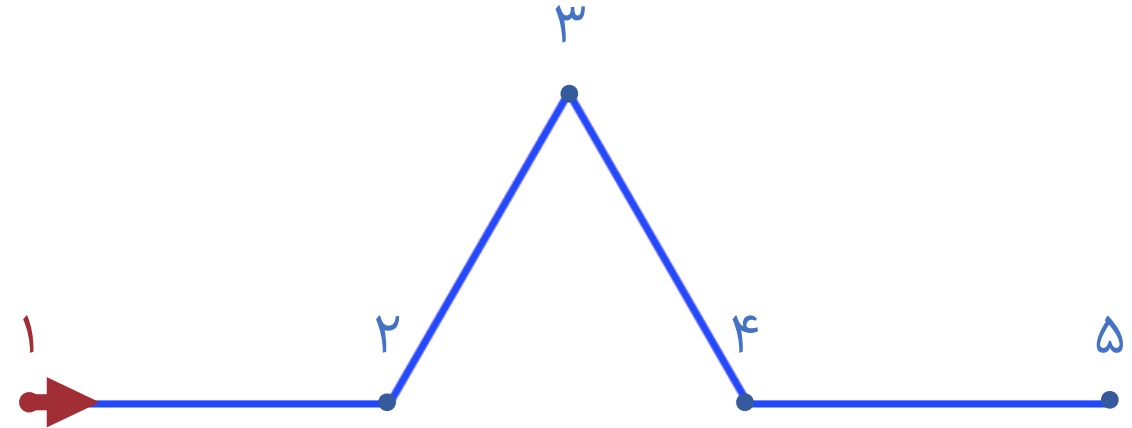




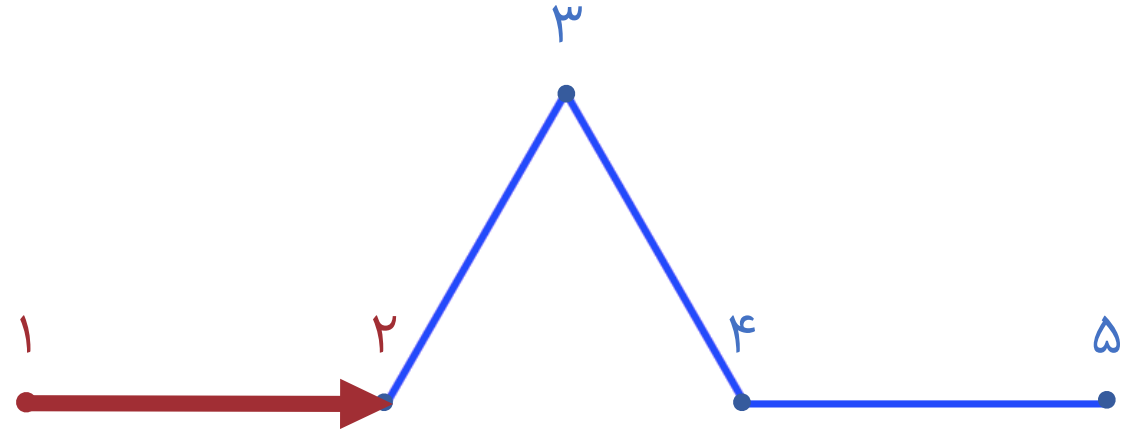




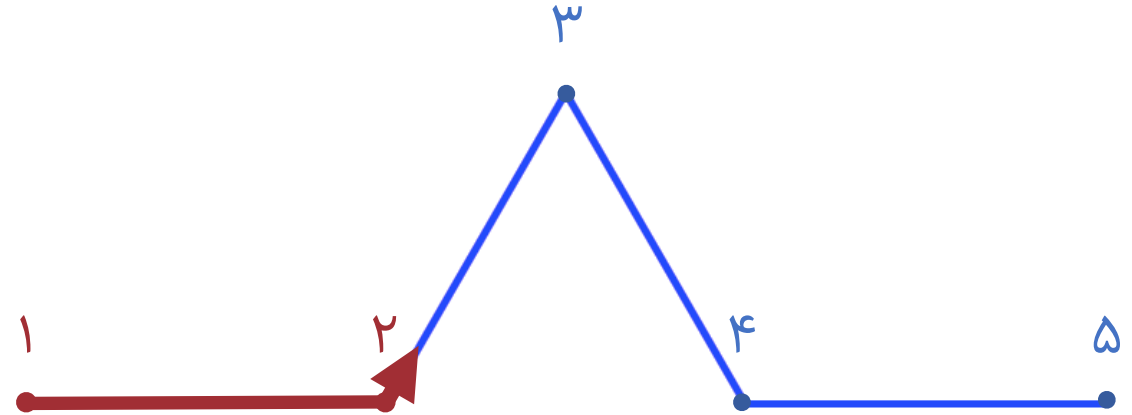
```
def draw(size):
```



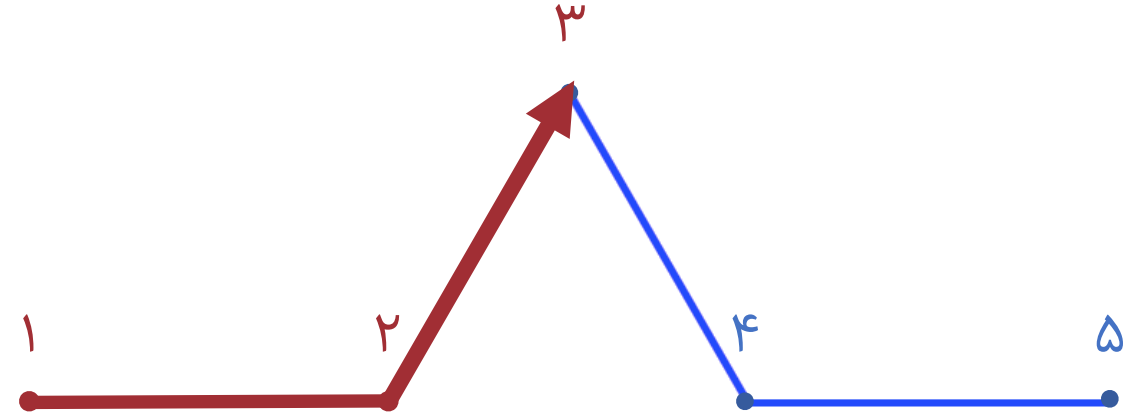
```
def draw(size):  
    forward(size/3)
```



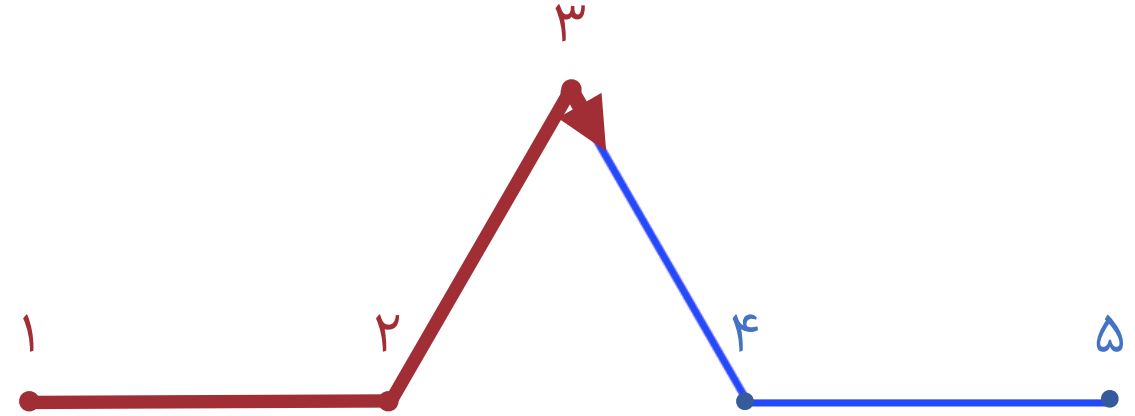

```
def draw(size):  
    forward(size/3)  
    left(60)
```



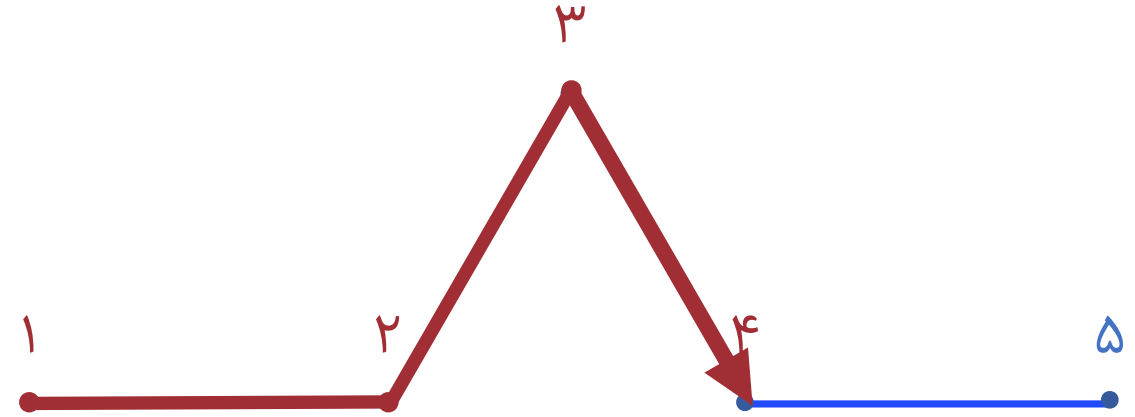
```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)
```



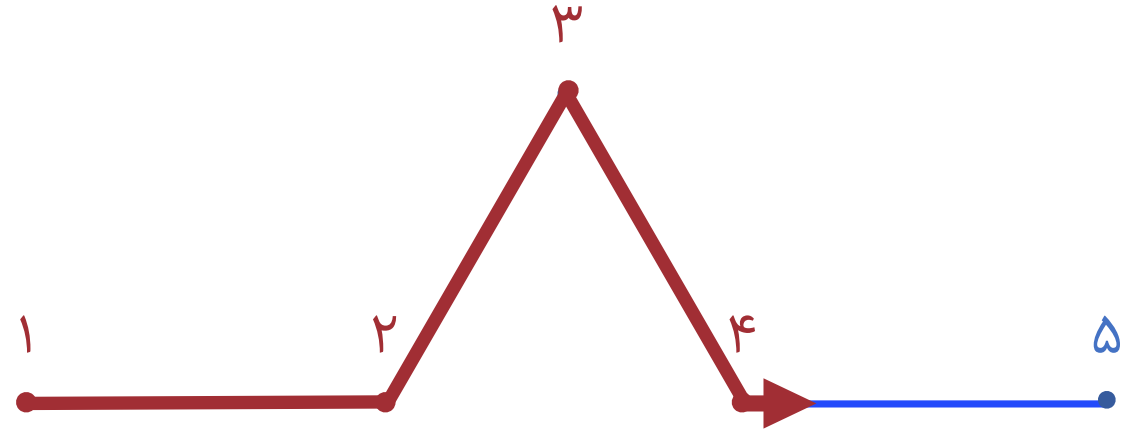
```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)  
    right(120)
```



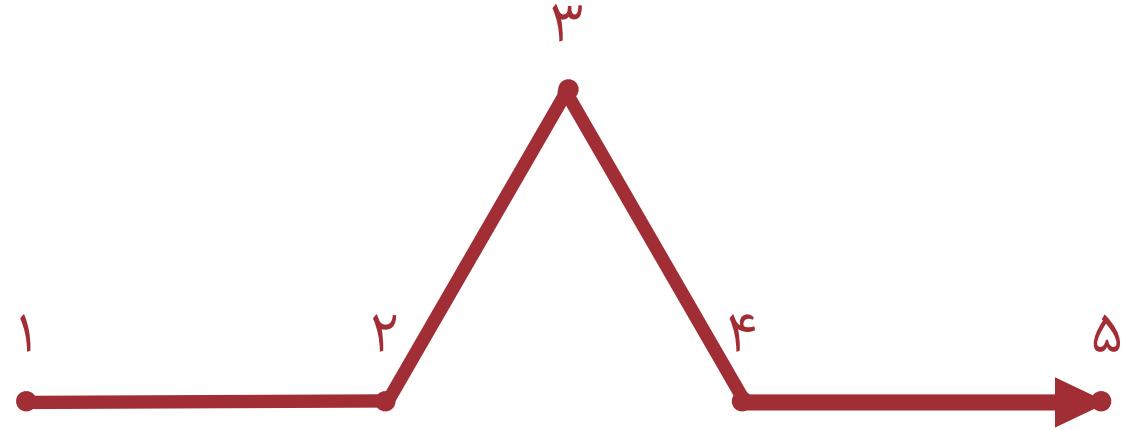
```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)  
    right(120)  
    forward(size/3)
```



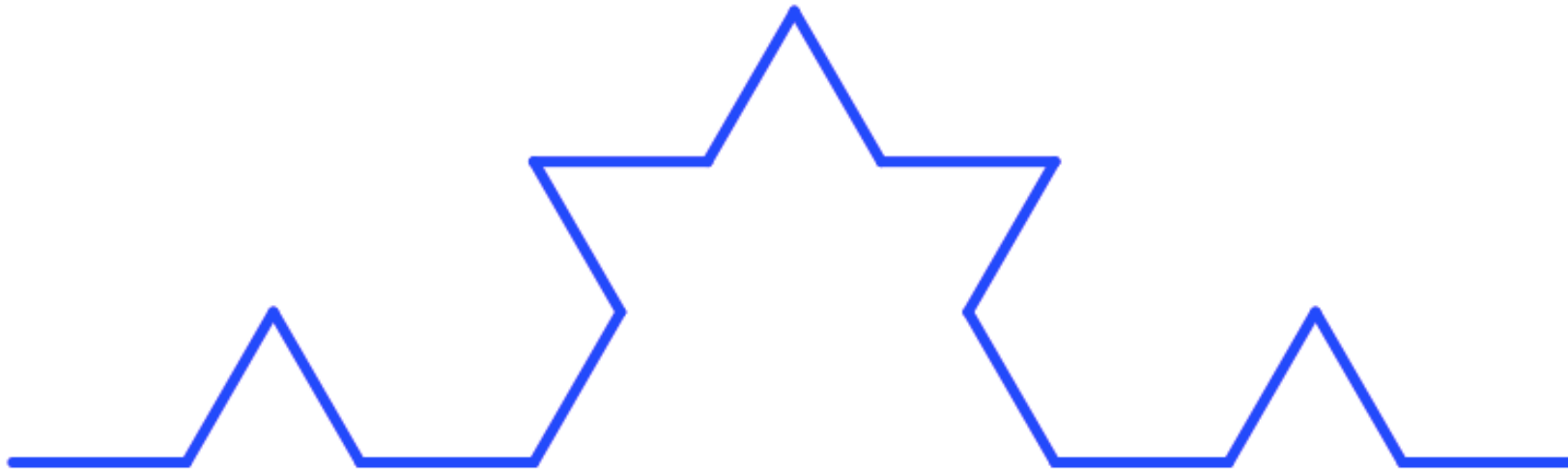
```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)  
    right(120)  
    forward(size/3)  
    left(60)
```



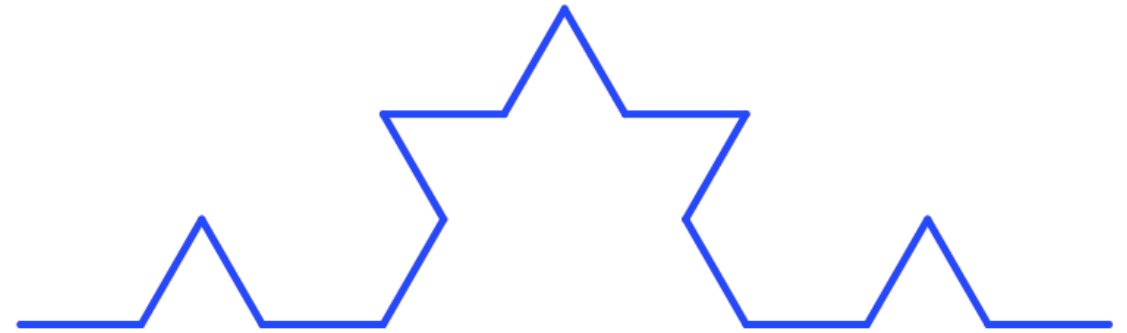
```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)  
    right(120)  
    forward(size/3)  
    left(60)  
    forward(size/3)
```



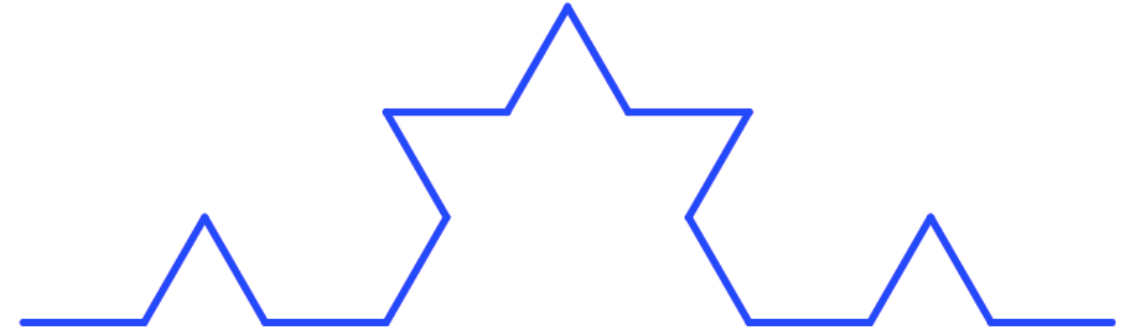
وقت کدزدن (:




```
def draw(size):  
    forward(size/3)  
    left(60)  
    forward(size/3)  
    right(120)  
    forward(size/3)  
    left(60)  
    forward(size/3)
```



```
def draw(size):  
    draw(size/3)  
    left(60)  
    draw(size/3)  
    right(120)  
    draw(size/3)  
    left(60)  
    draw(size/3)
```



```
def draw(size, level):
```

```
    draw(size/3)
```

```
    left(60)
```

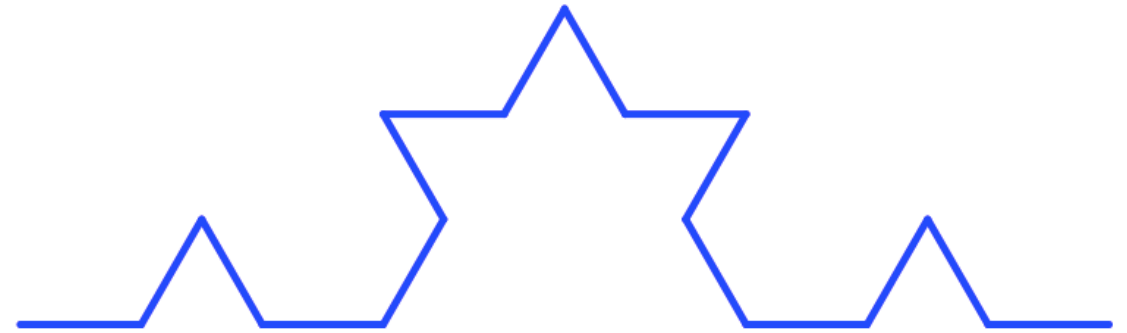
```
    draw(size/3)
```

```
    right(120)
```

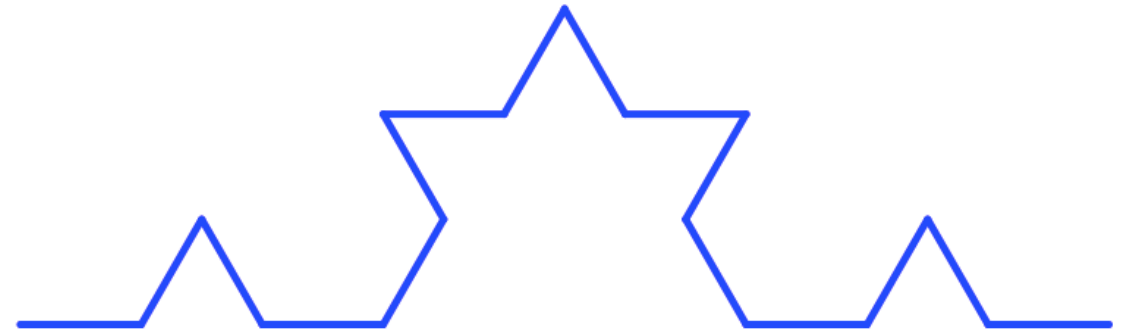
```
    draw(size/3)
```

```
    left(60)
```

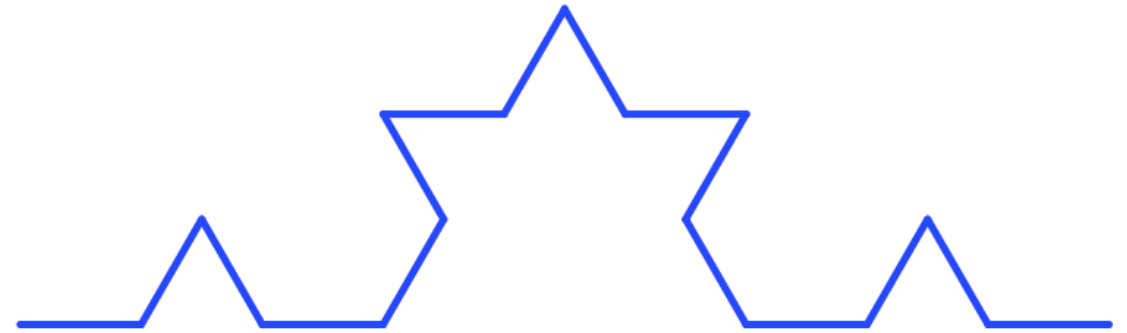
```
    draw(size/3)
```



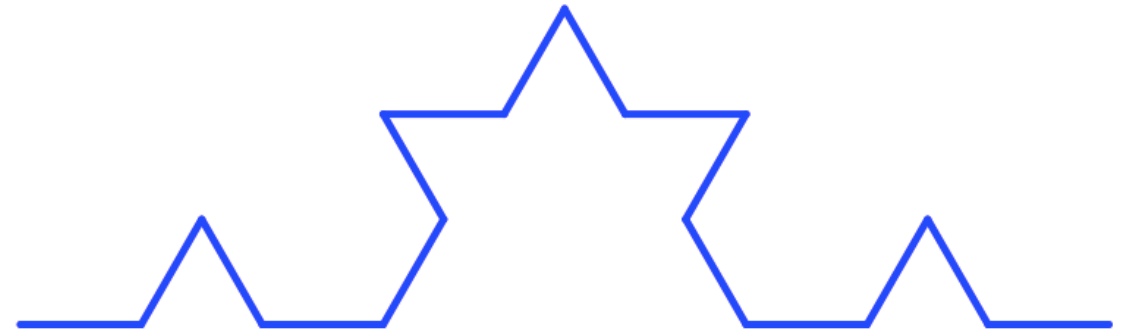
```
def draw(size, level):  
    if level==0:  
  
        draw(size/3)  
        left(60)  
        draw(size/3)  
        right(120)  
        draw(size/3)  
        left(60)  
        draw(size/3)
```



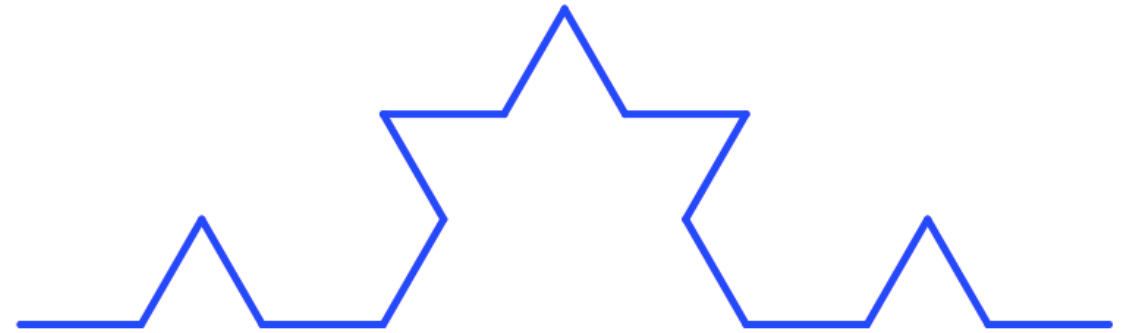
```
def draw(size, level):  
    if level==0:  
        forward(size)  
  
    draw(size/3)  
    left(60)  
    draw(size/3)  
    right(120)  
    draw(size/3)  
    left(60)  
    draw(size/3)
```



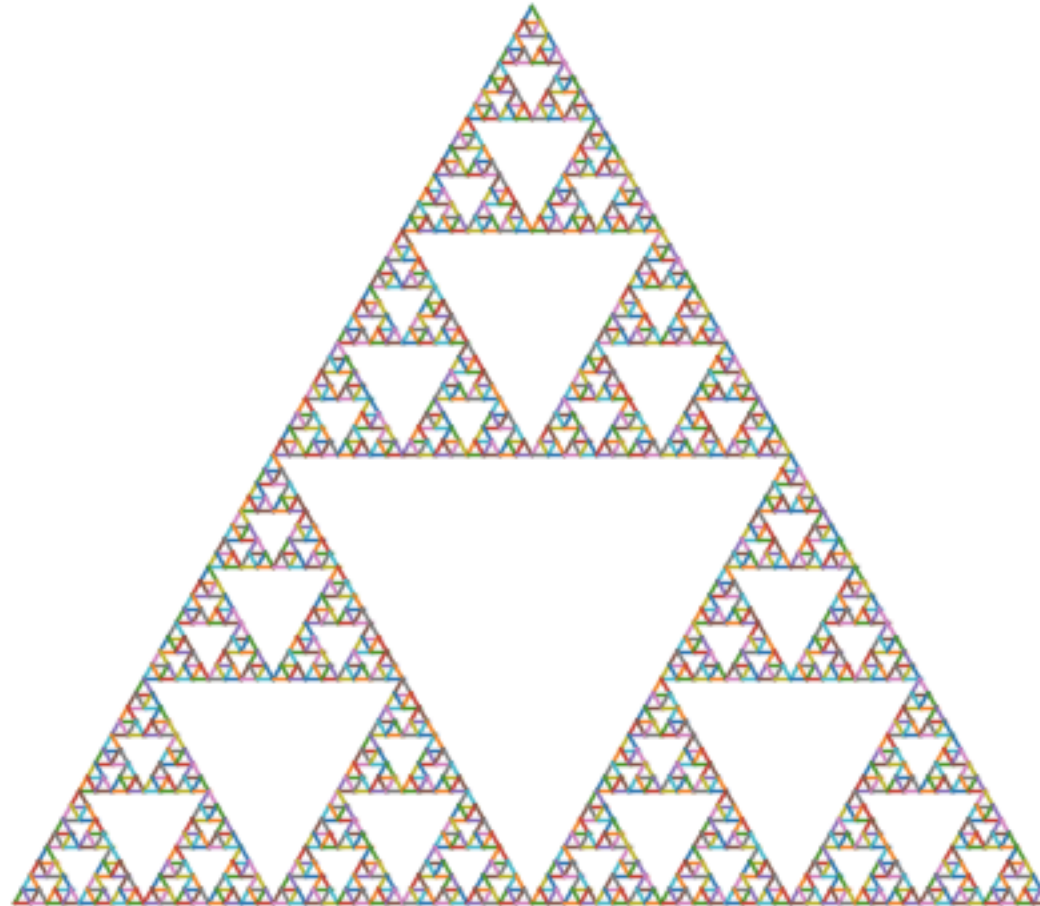
```
def draw(size, level):  
    if level==0:  
        forward(size)  
        return  
    draw(size/3)  
    left(60)  
    draw(size/3)  
    right(120)  
    draw(size/3)  
    left(60)  
    draw(size/3)
```

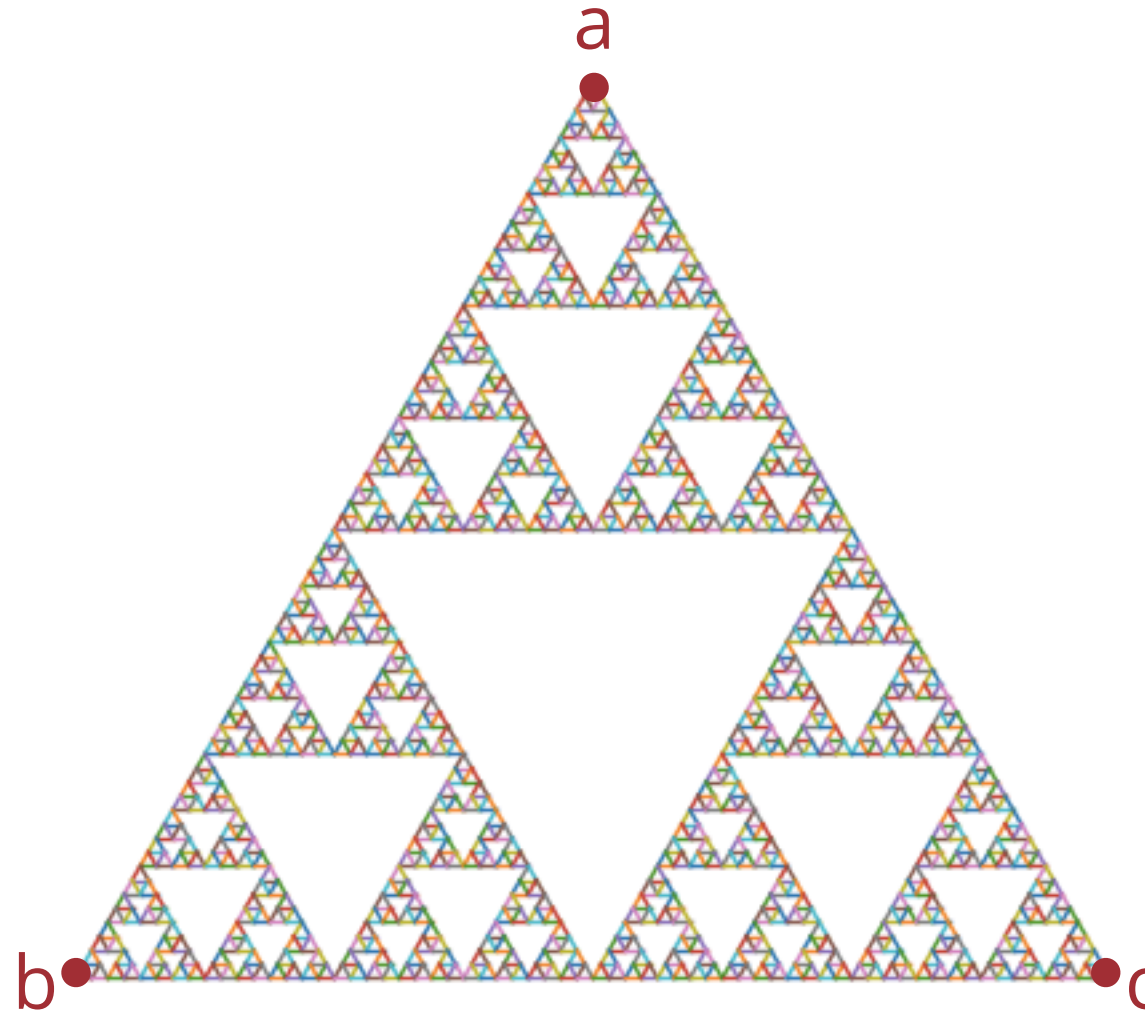


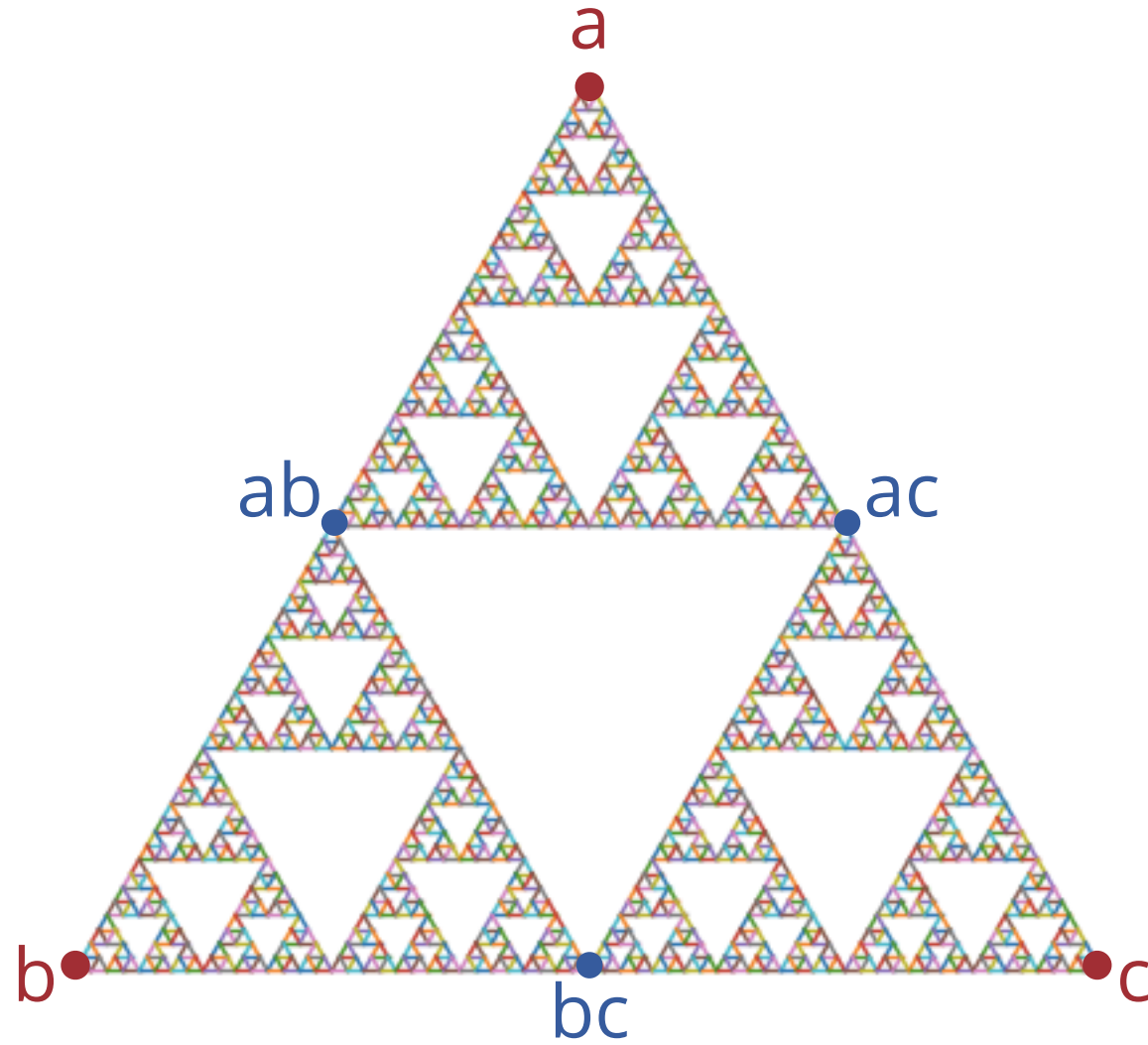
```
def draw(size, level):  
    if level==0:  
        forward(size)  
        return  
    draw(size/3, level-1)  
    left(60)  
    draw(size/3, level-1)  
    right(120)  
    draw(size/3, level-1)  
    left(60)  
    draw(size/3, level-1)
```

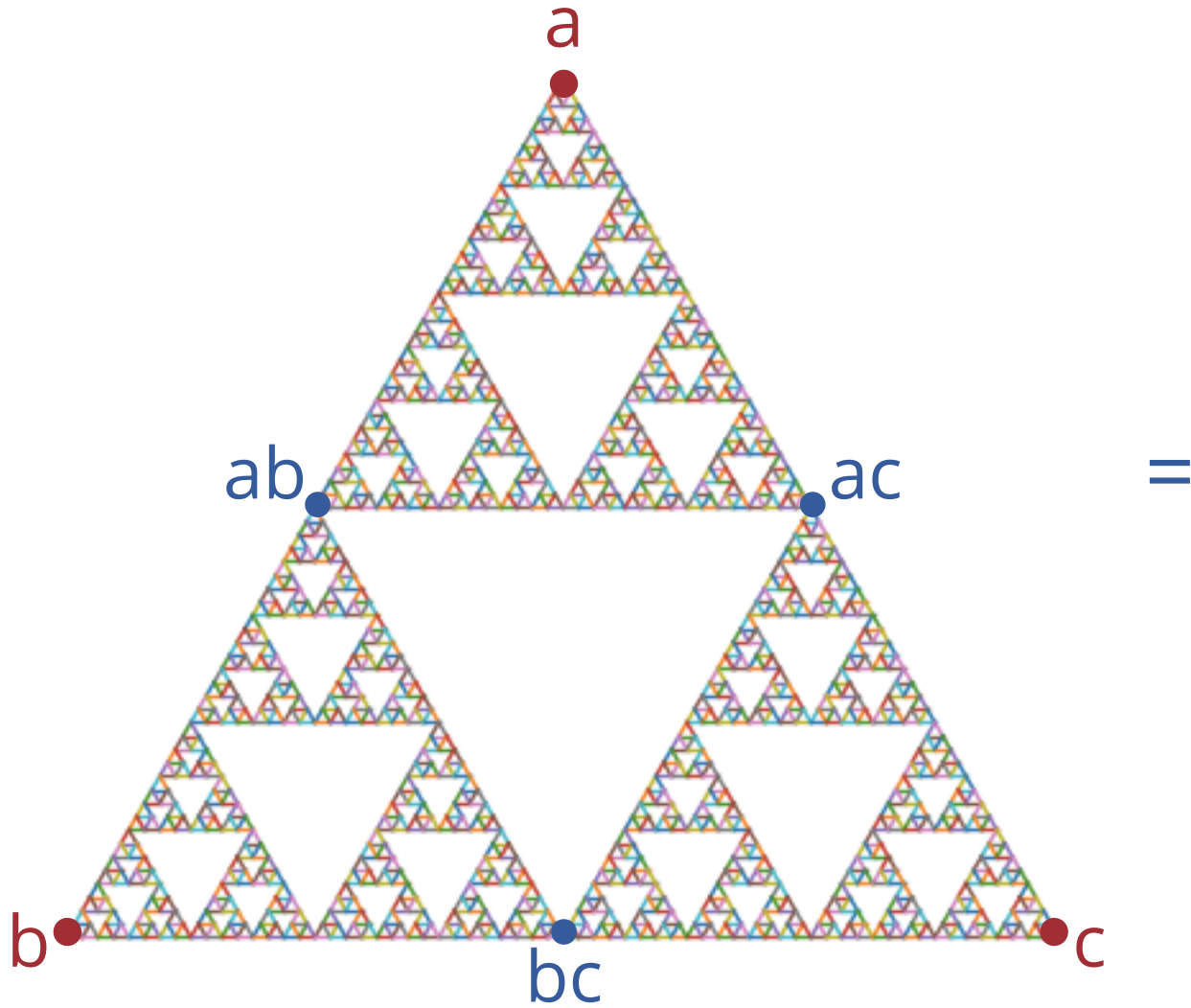


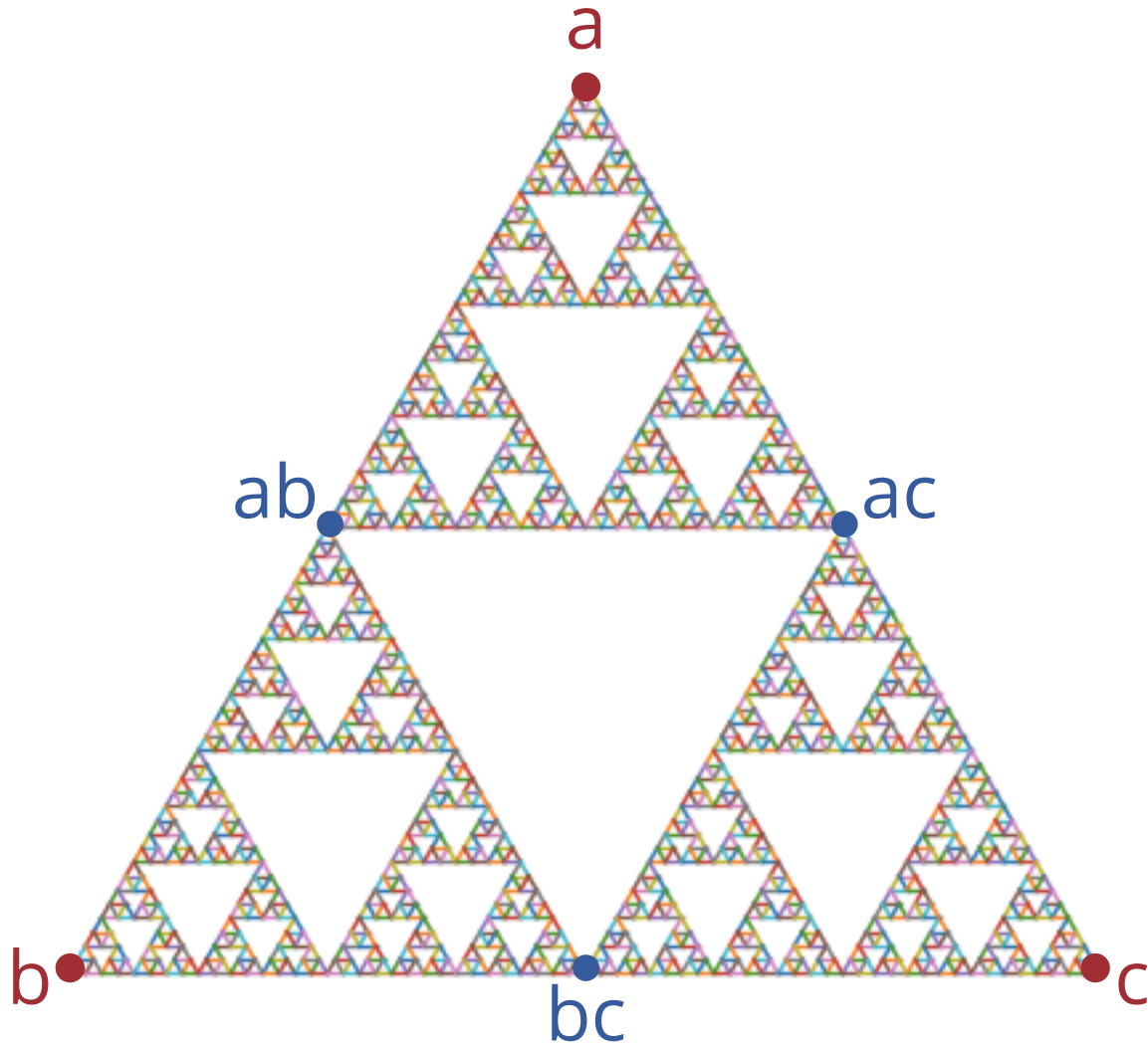
وقت کدزدن (:



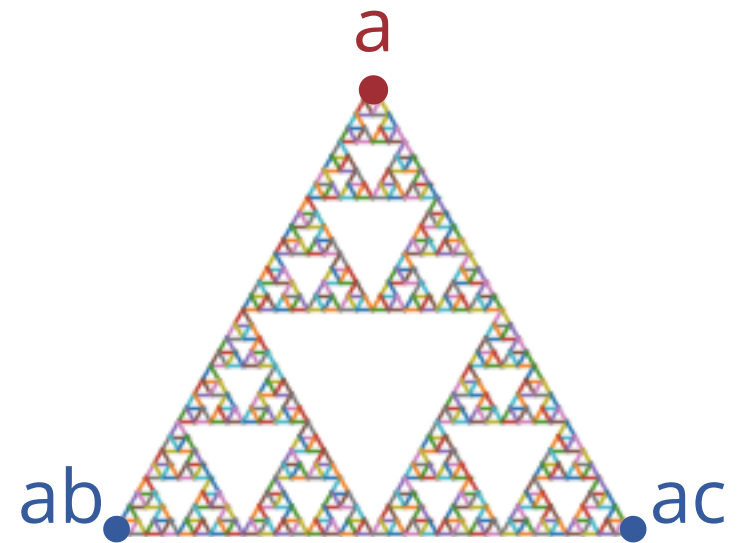


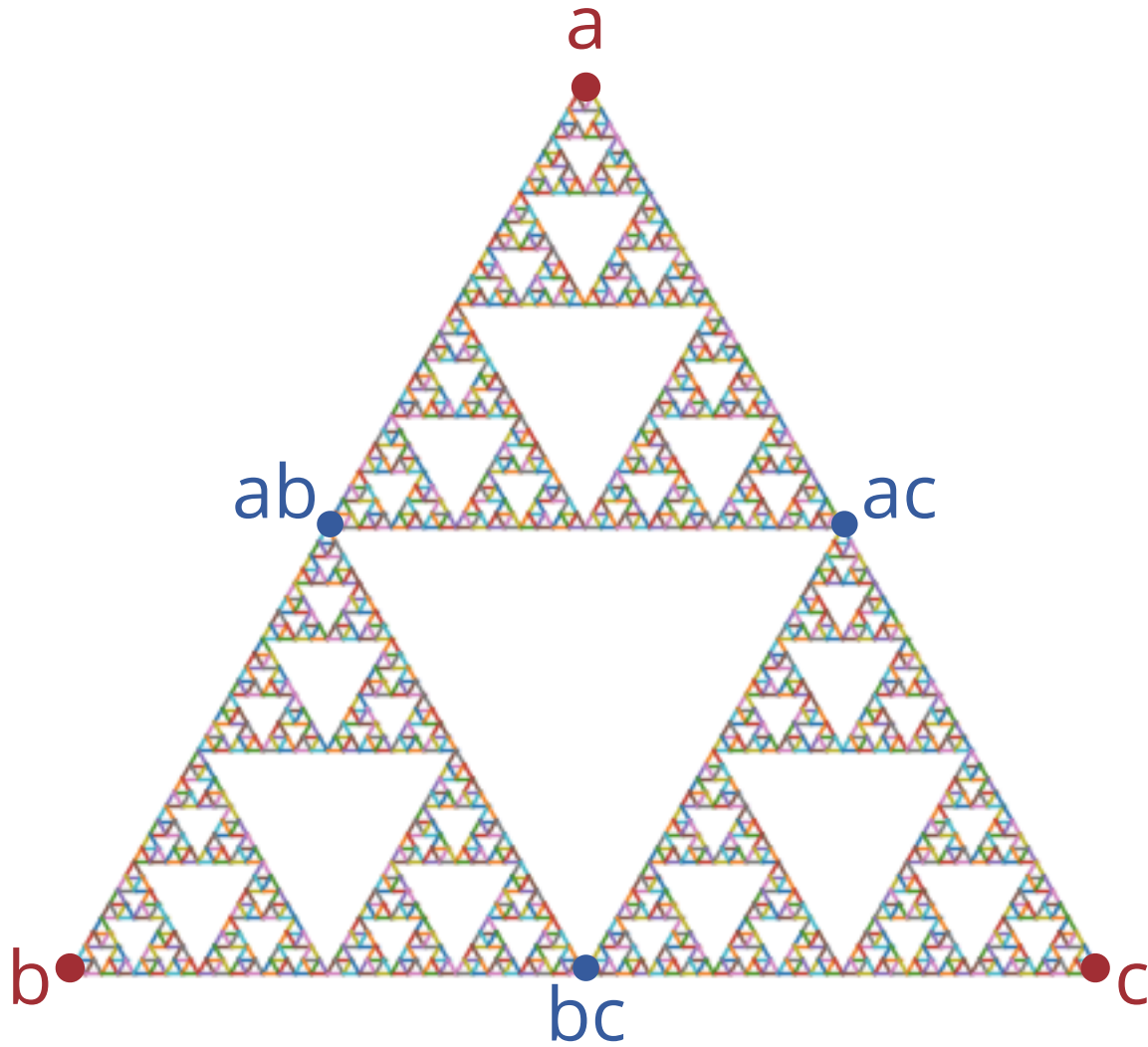




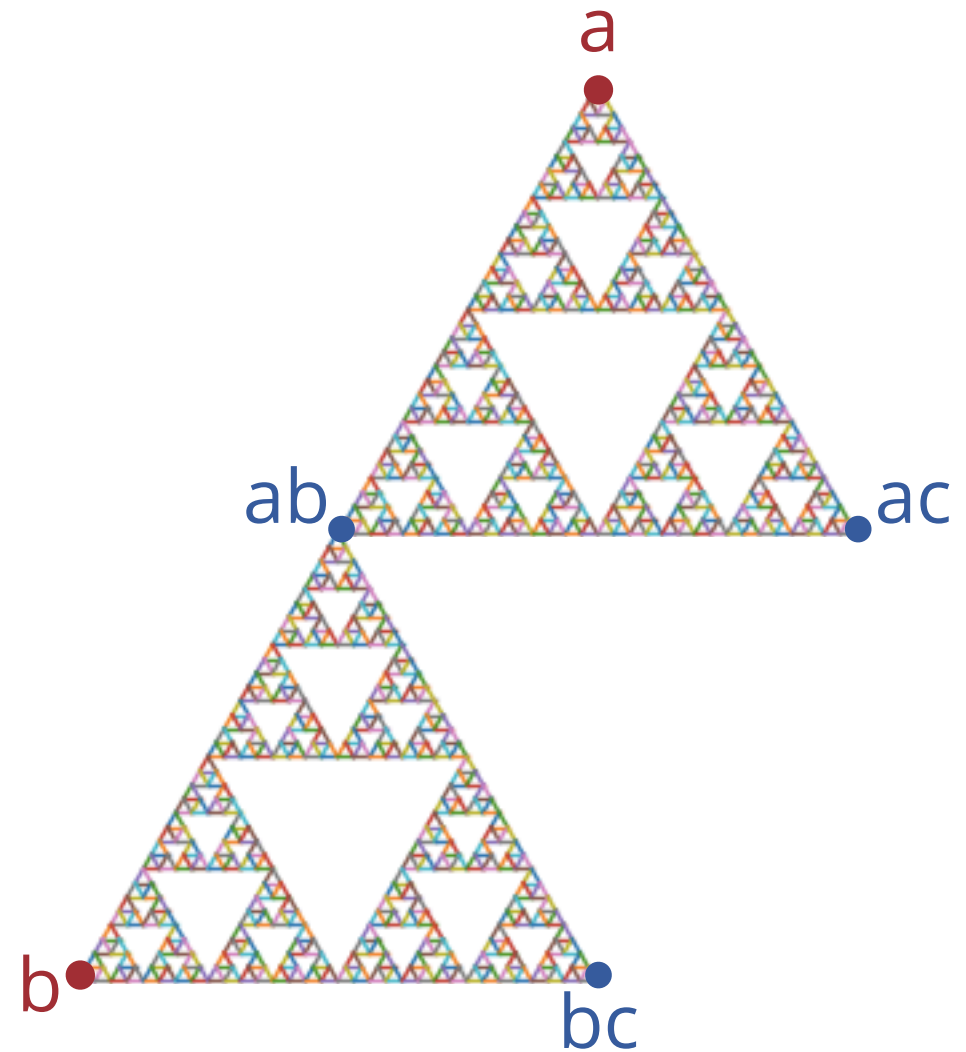


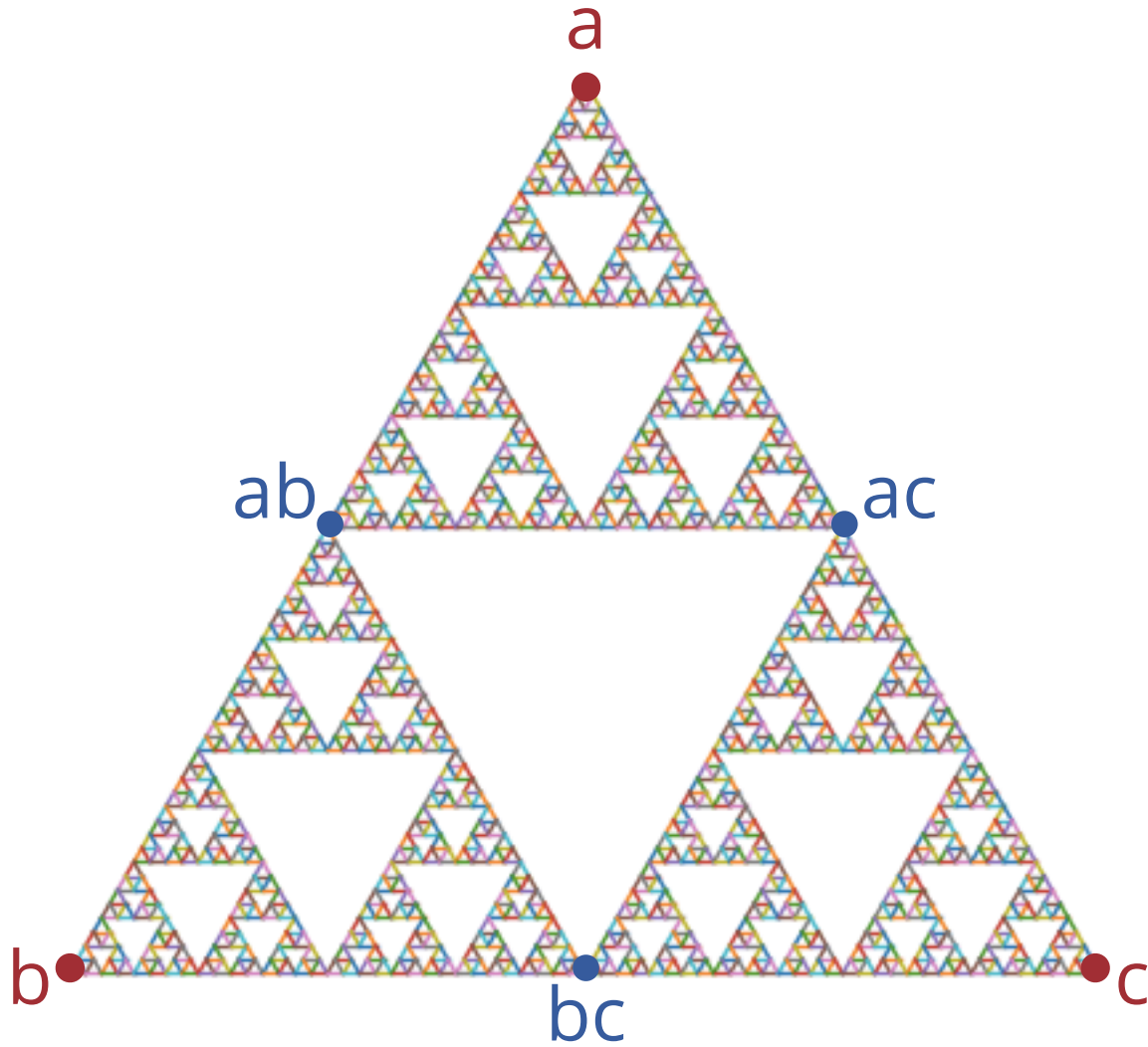
=



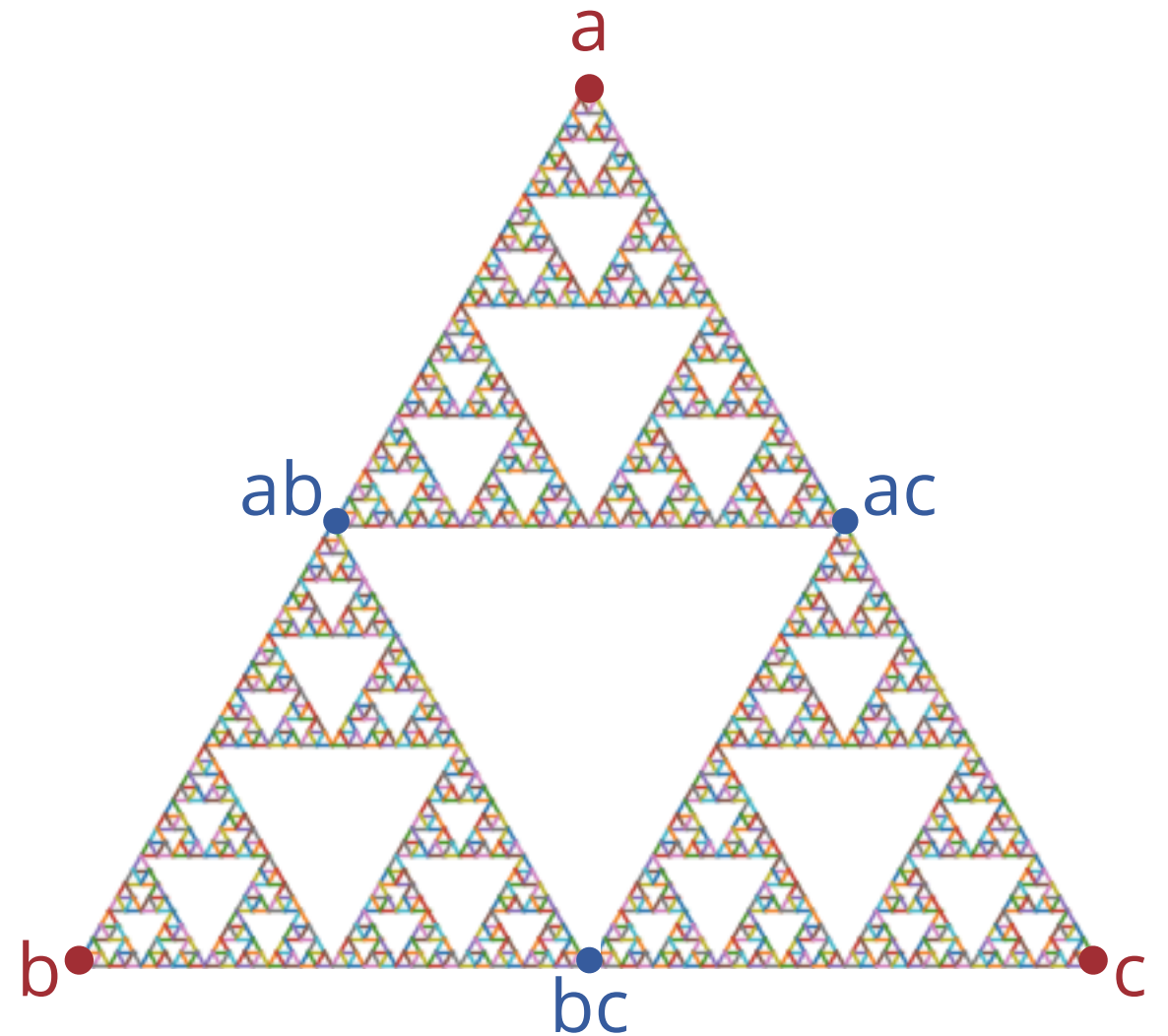


=





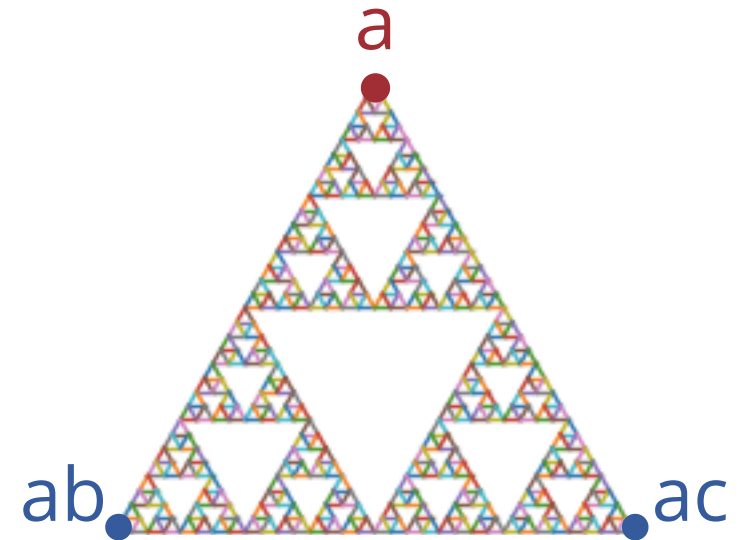
=



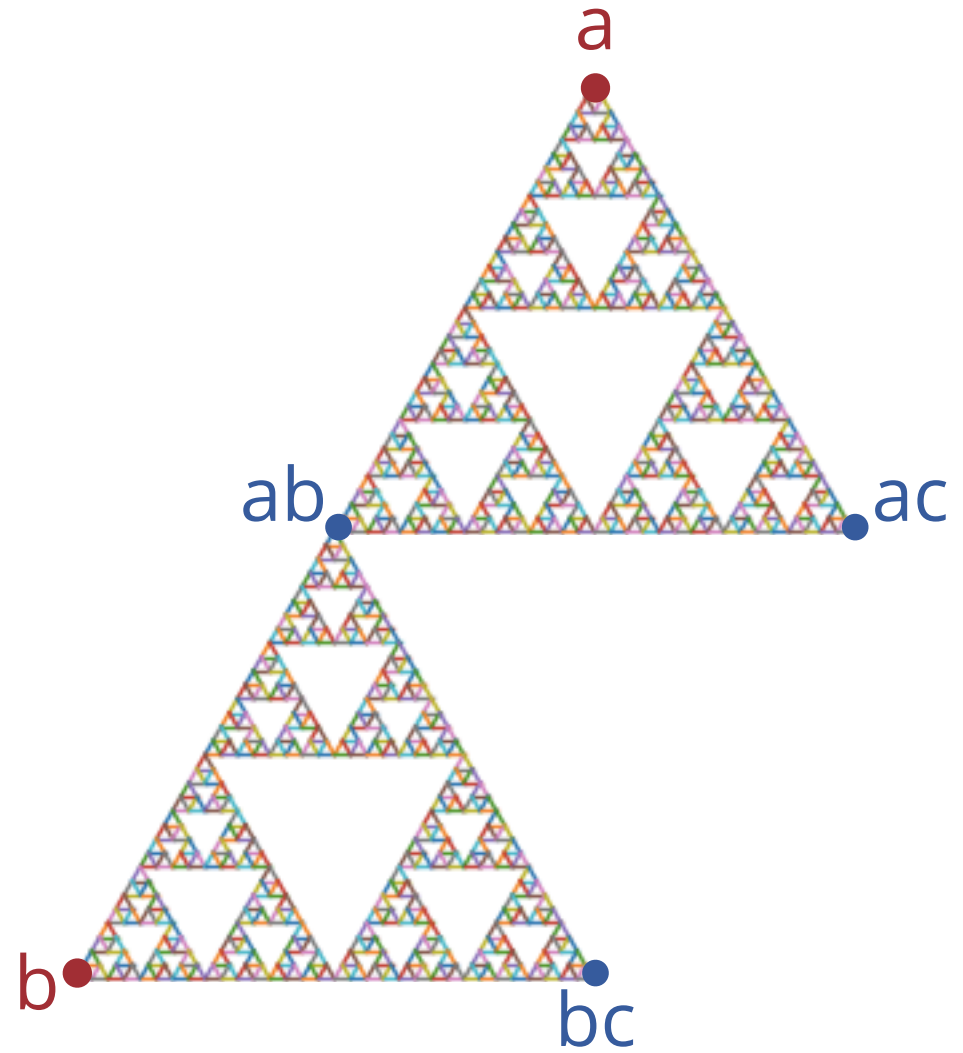
```
def draw(a, b, c):
```



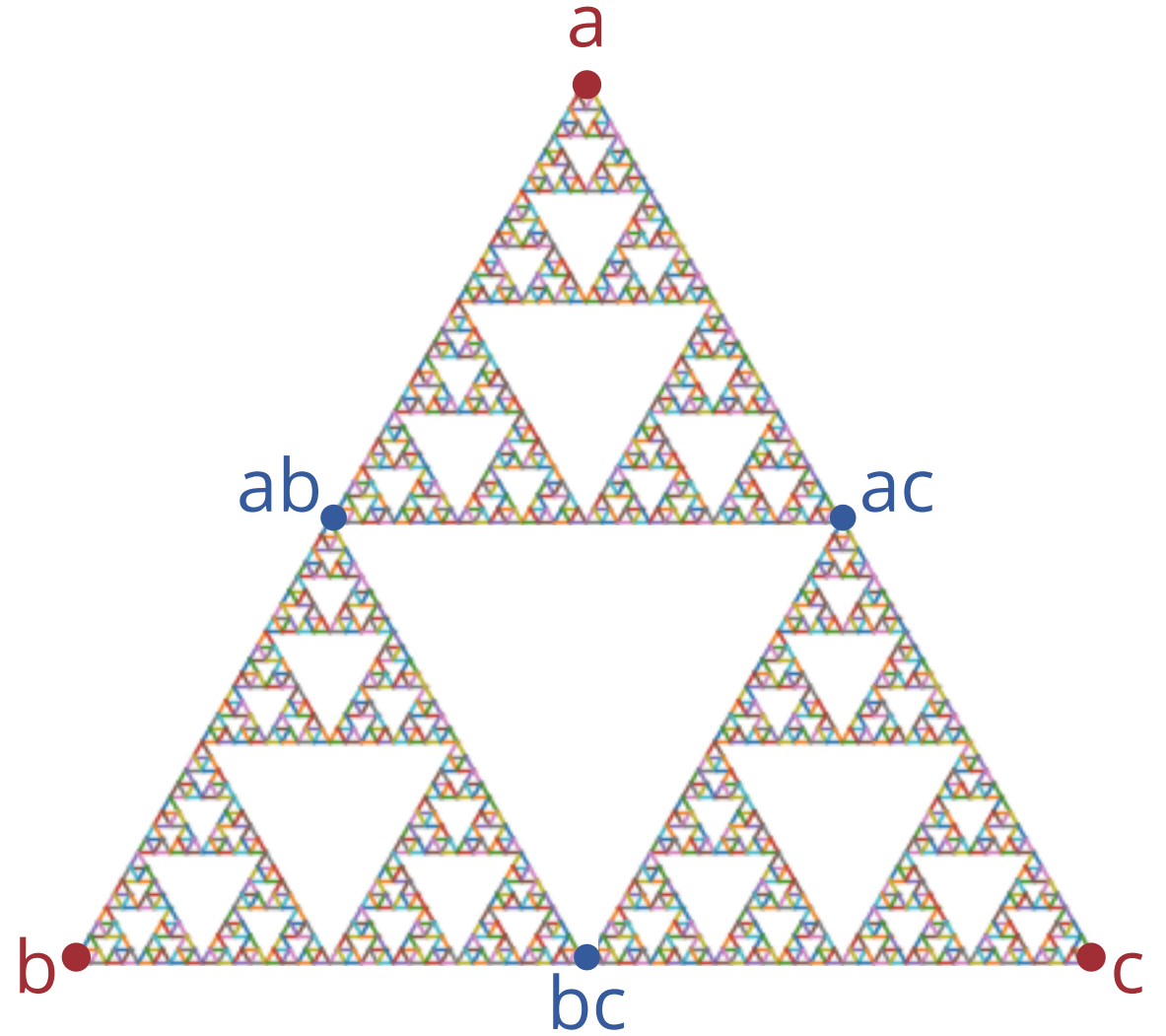
```
def draw(a, b, c):  
    draw(a, ab, ac)
```



```
def draw(a, b, c):  
    draw(a, ab, ac)  
    draw(ab, b, bc)
```



```
def draw(a, b, c):  
    draw(a, ab, ac)  
    draw(ab, b, bc)  
    draw(ac, bc, c)
```

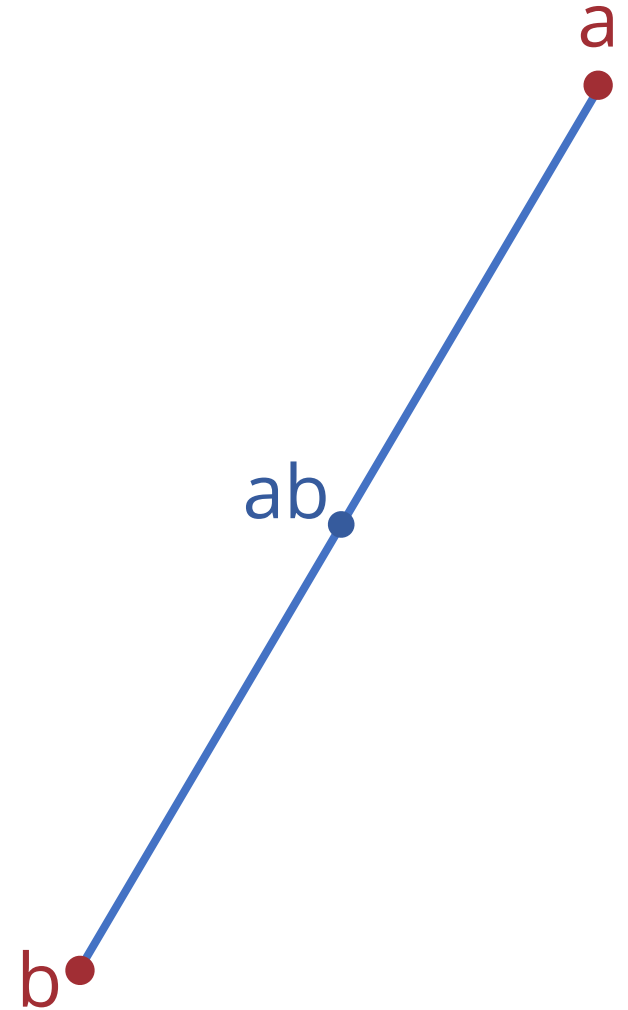


```
def draw(a, b, c):
```

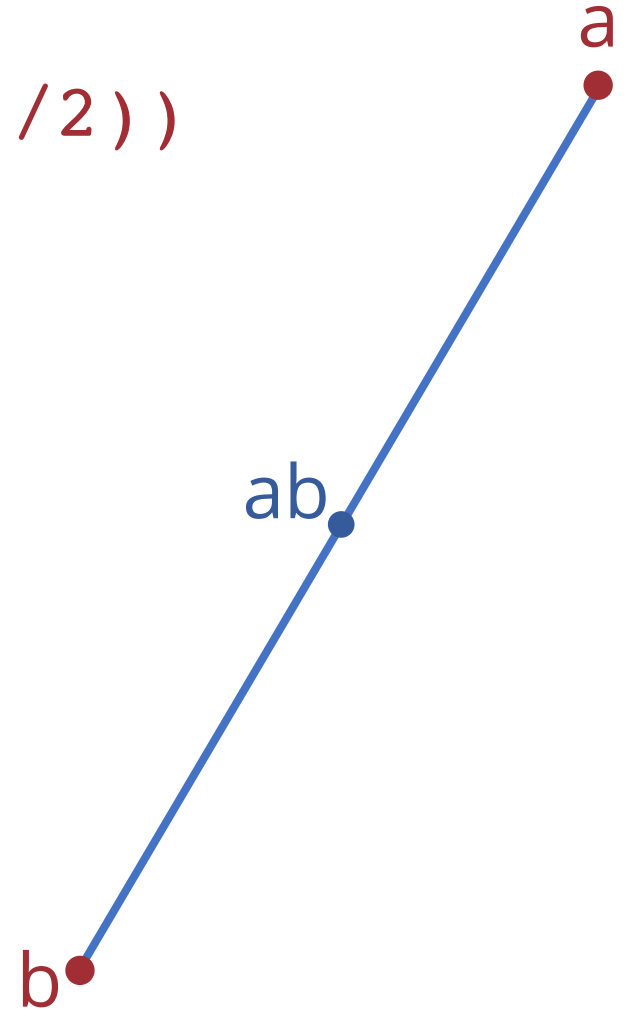
```
    draw(a, ab, ac)
```

```
    draw(ab, b, bc)
```

```
    draw(ac, bc, c)
```



```
def draw(a, b, c):  
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
  
    draw(a, ab, ac)  
    draw(ab, b, bc)  
    draw(ac, bc, c)
```



```
def draw(a, b, c):  
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
    ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))  
  
    draw(a, ab, ac)  
    draw(ab, b, bc)  
    draw(ac, bc, c)
```

```
def draw(a, b, c):  
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
    ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))  
    bc = ((b[0]+c[0])/2, (b[1]+c[1])/2))  
    draw(a, ab, ac)  
    draw(ab, b, bc)  
    draw(ac, bc, c)
```

```
def draw(a, b, c, level):
```

```
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))
```

```
    ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))
```

```
    bc = ((b[0]+c[0])/2, (b[1]+c[1])/2))
```

```
    draw(a, ab, ac)
```

```
    draw(ab, b, bc)
```

```
    draw(ac, bc, c)
```



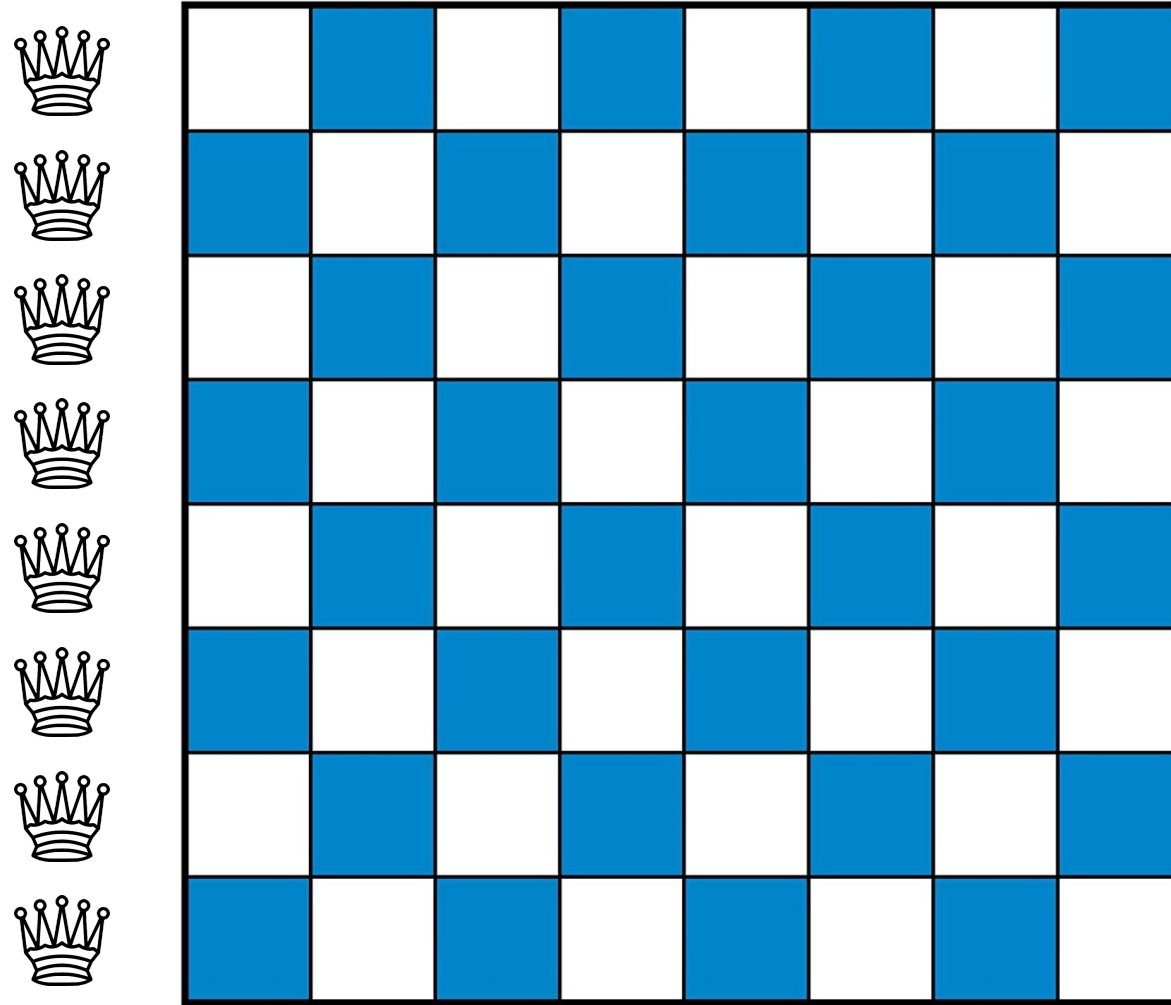
```
def draw(a, b, c, level):  
    if level==0:  
  
        ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
        ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))  
        bc = ((b[0]+c[0])/2, (b[1]+c[1])/2))  
        draw(a, ab, ac)  
        draw(ab, b, bc)  
        draw(ac, bc, c)
```

```
def draw(a, b, c, level):  
    if level==0:  
        line(a, b)  
        line(a, c)  
        line(b, c)  
        return  
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
    ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))  
    bc = ((b[0]+c[0])/2, (b[1]+c[1])/2))  
    draw(a, ab, ac)  
    draw(ab, b, bc)  
    draw(ac, bc, c)
```

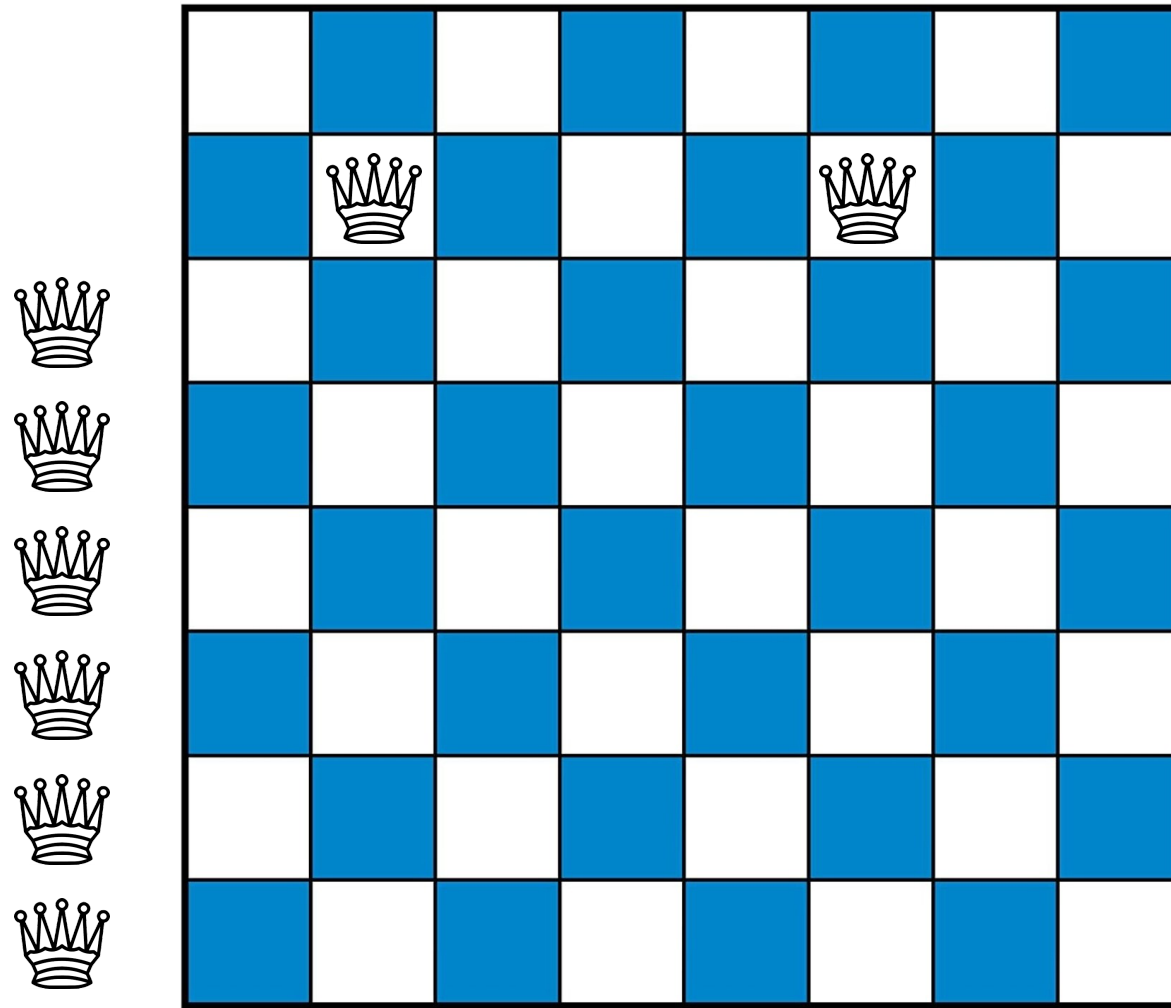
```
def draw(a, b, c, level):  
    if level==0:  
        line(a, b)  
        line(a, c)  
        line(b, c)  
        return  
    ab = ((a[0]+b[0])/2, (a[1]+b[1])/2))  
    ac = ((a[0]+c[0])/2, (a[1]+c[1])/2))  
    bc = ((b[0]+c[0])/2, (b[1]+c[1])/2))  
    draw(a, ab, ac, level-1)  
    draw(ab, b, bc, level-1)  
    draw(ac, bc, c, level-1)
```

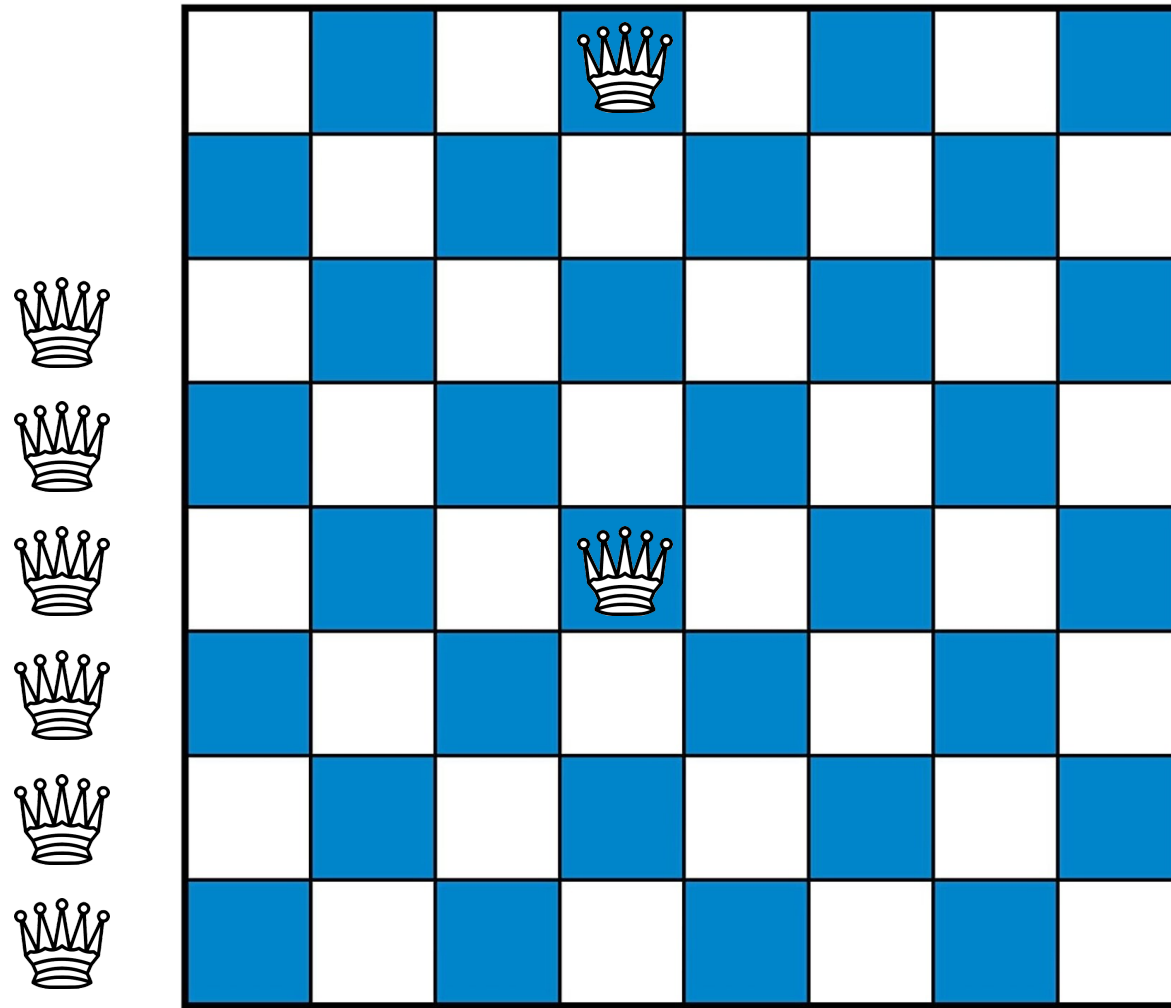
وقت کدزدن (:

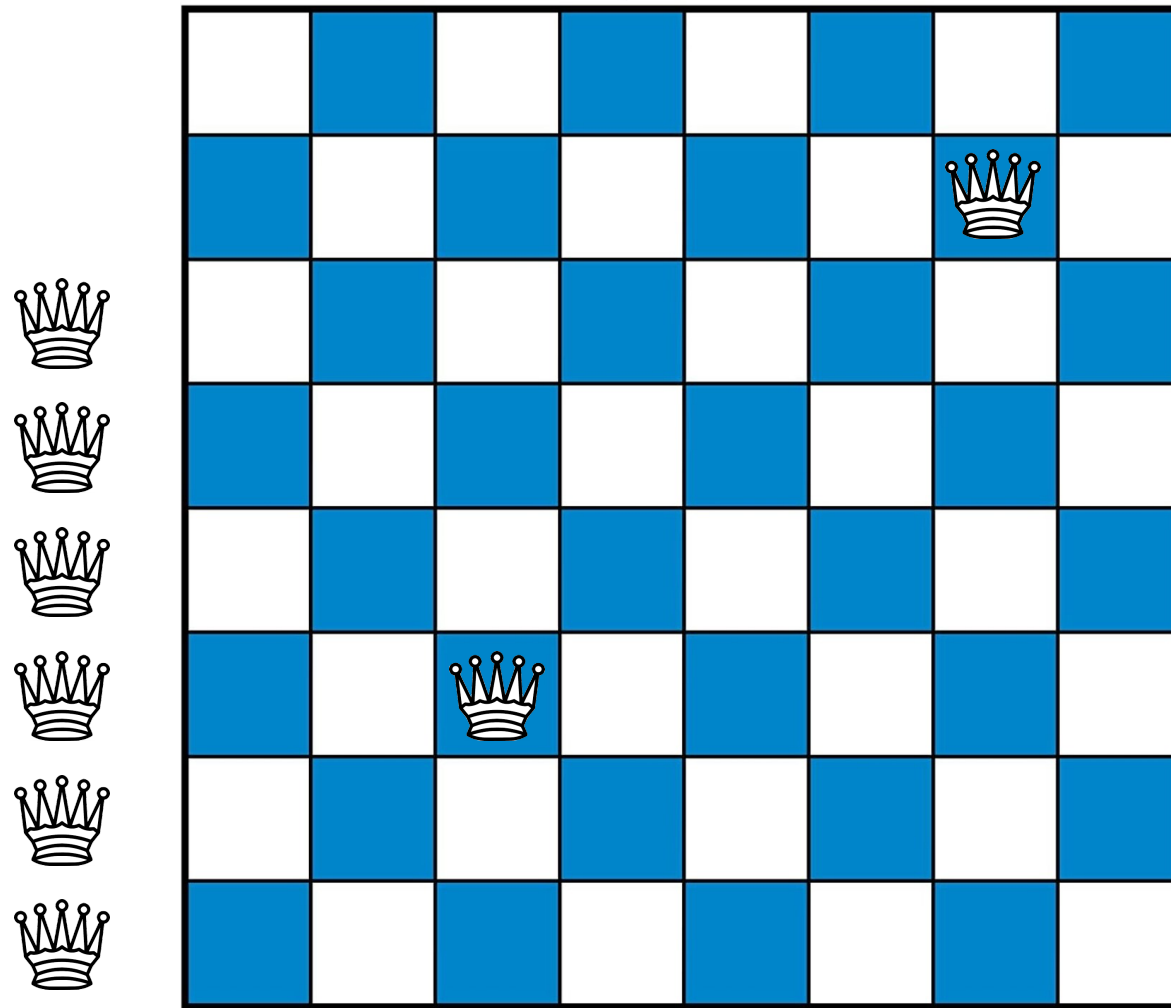
۸ وزیر را در صفحه‌ی شطرنج بگذاریم طوری که هیچ‌یک دیگری را تهدید نکند



دو وزیر یکدیگر را تهدید می‌کنند اگر در یک سطر، ...







گام نخست: چک کردن همه‌ی جایگشت‌های ممکن

```
for i in range(8):
```

گام نخست: چک کردن همه‌ی جایگشت‌های ممکن

```
for i in range(8):  
    for j in range(8):  
        ...
```

گام نخست: چک کردن همه‌ی جایگشت‌های ممکن

```
for i1 in range(8):  
    for i2 in range(8):  
        for i3 in range(8):  
            for i4 in range(8):  
                for i5 in range(8):  
                    for i6 in range(8):  
                        for i7 in range(8):  
                            for i8 in range(8):
```

گام نخست: چک کردن همه‌ی جایگشت‌های ممکن

```
for i1 in range(8):  
    for i2 in range(8):  
        for i3 in range(8):  
            for i4 in range(8):  
                for i5 in range(8):  
                    for i6 in range(8):  
                        for i7 in range(8):  
                            for i8 in range(8):  
                                if i1 != i2 and i1 != i3 and..
```

```
def find(row):  
    for i in range(8):
```

```
def find(row):  
    for i in range(8):  
        if not col[i]:
```

```
def find(row):  
    for i in range(8):  
        if not col[i]:  
            queens[row] = i  
            col[i] = True
```

```
def find(row):  
    for i in range(8):  
        if not col[i]:  
            queens[row] = i  
            col[i] = True  
            find(row+1)
```



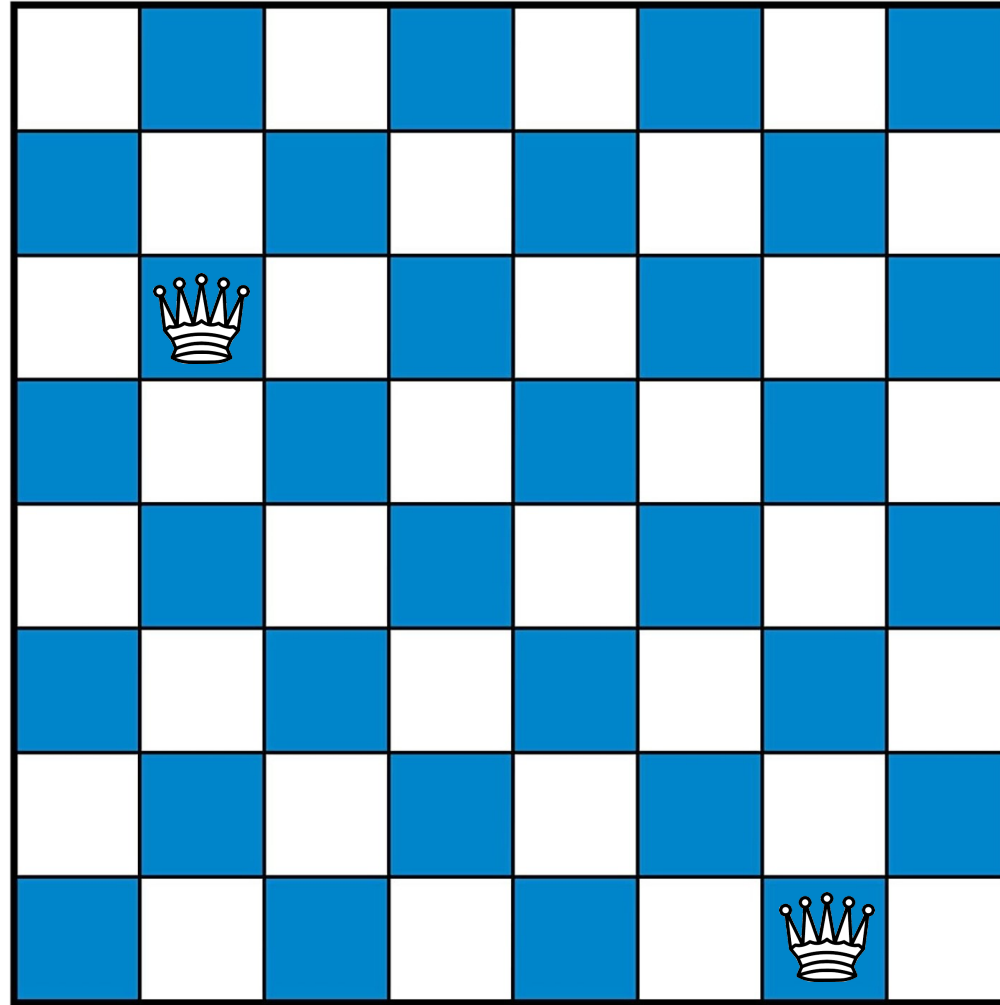
```
def find(row):  
    for i in range(8):  
        if not col[i]:  
            queens[row] = i  
            col[i] = True  
            find(row+1)  
            col[i] = False
```

```
def find(row):  
  
    for i in range(8):  
        if not col[i]:  
            queens[row] = i  
            col[i] = True  
            find(row+1)  
            col[i] = False
```

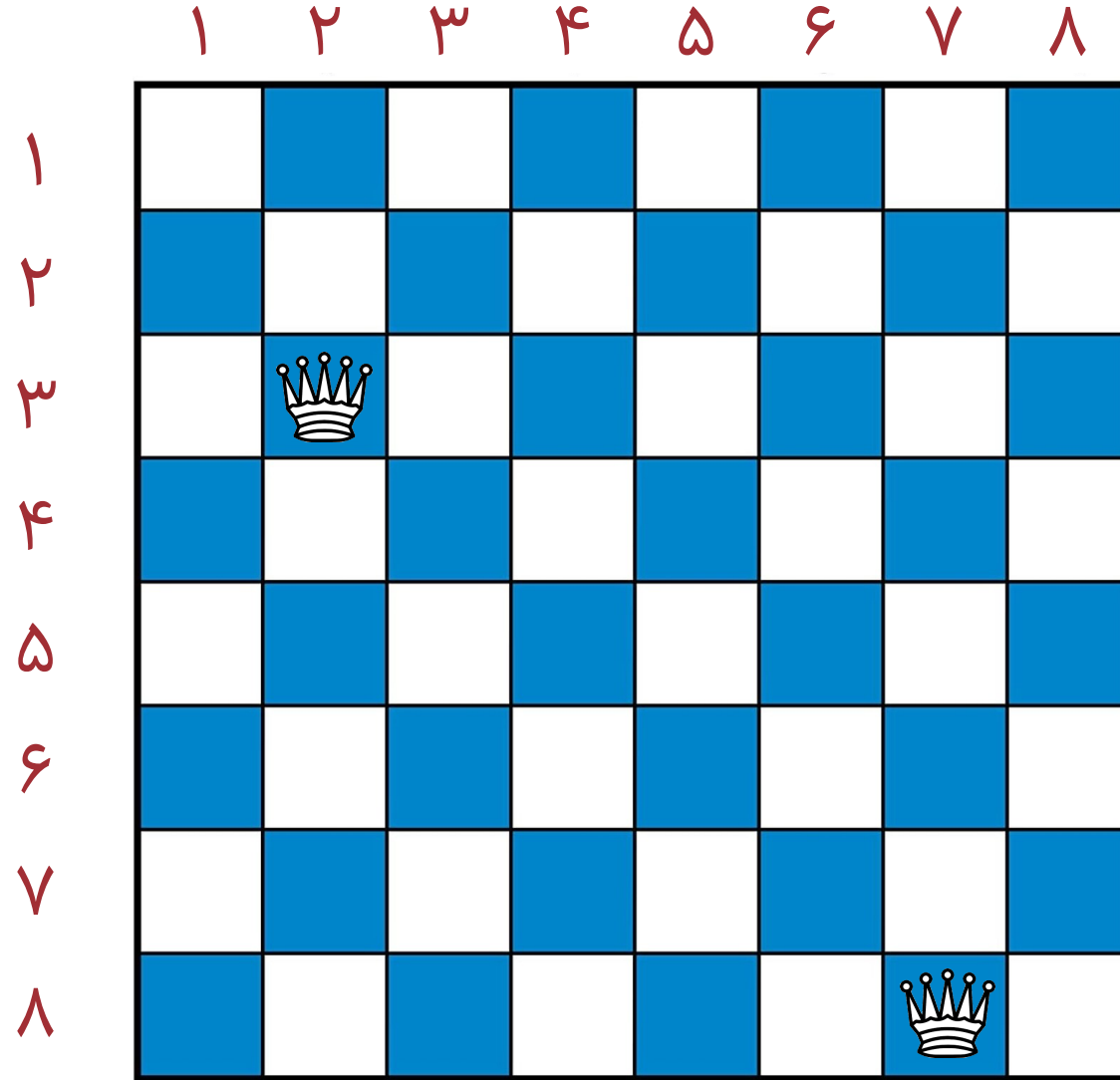
```
def find(row):  
    if row == 8:  
        print(queens)  
        return  
    for i in range(8):  
        if not col[i]:  
            queens[row] = i  
            col[i] = True  
            find(row+1)  
            col[i] = False
```

وقت کدزدن (:

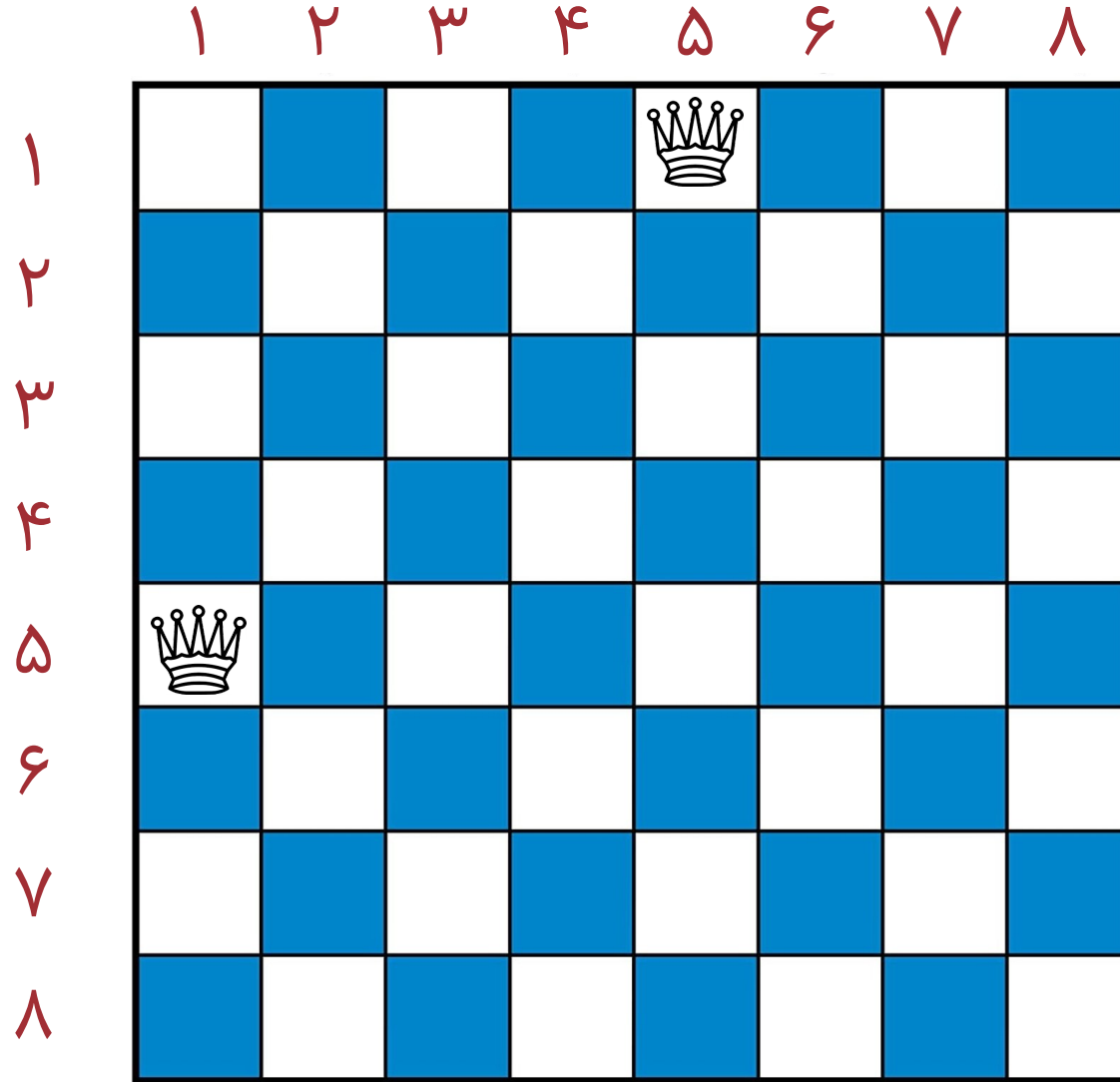
گام دوم: کنترل قطرها



گام دوم: کنترل قطرها



گام دوم: کنترل قطرها



```
def find(row):  
    if row == 8:  
        print_solution(queens)  
        return  
    for i in range(8):  
        if put(row, i):  
            find(row+1)  
            pick(row, i)
```



```
def put(r, c):  
    if col[c] or diag1[r+c] or diag2[r-c+7]:  
        return False
```

```
def put(r, c):  
    if col[c] or diag1[r+c] or diag2[r-c+7]:  
        return False  
    col[c] = diag1[r+c] = diag2[r-c+7] = True  
    queens[r] = c
```

```
def put(r, c):  
    if col[c] or diag1[r+c] or diag2[r-c+7]:  
        return False  
    col[c] = diag1[r+c] = diag2[r-c+7] = True  
    queens[r] = c  
  
def pick(r, c):  
    col[c] = diag1[r+c] = diag2[r-c+7] = False
```

وقت کدزدن (:

شاد و تن درست باشید :-)