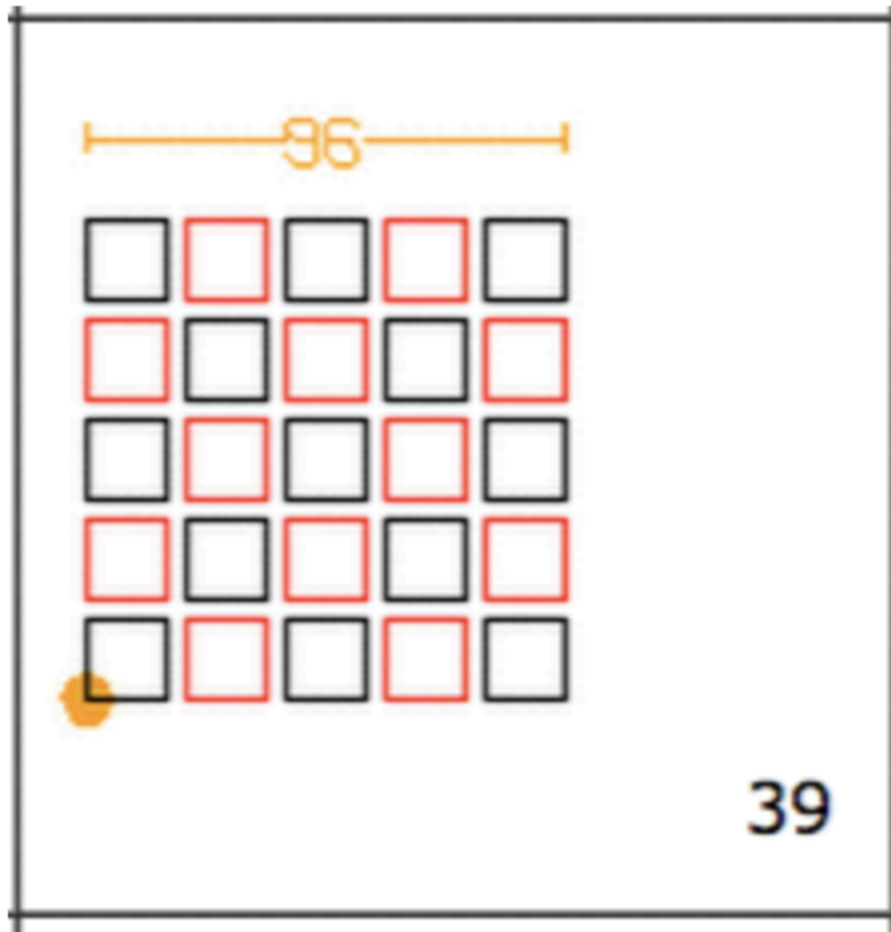


```
In [1]: from mplturtle import *
```

Version 0.0.2



```
In [6]: def square(size):
        for i in range(4):
            forward(size)
            right(90)

        def forward(distance, pen="down"):
            from mplturtle import forward as original_forward

            was_up=isup()

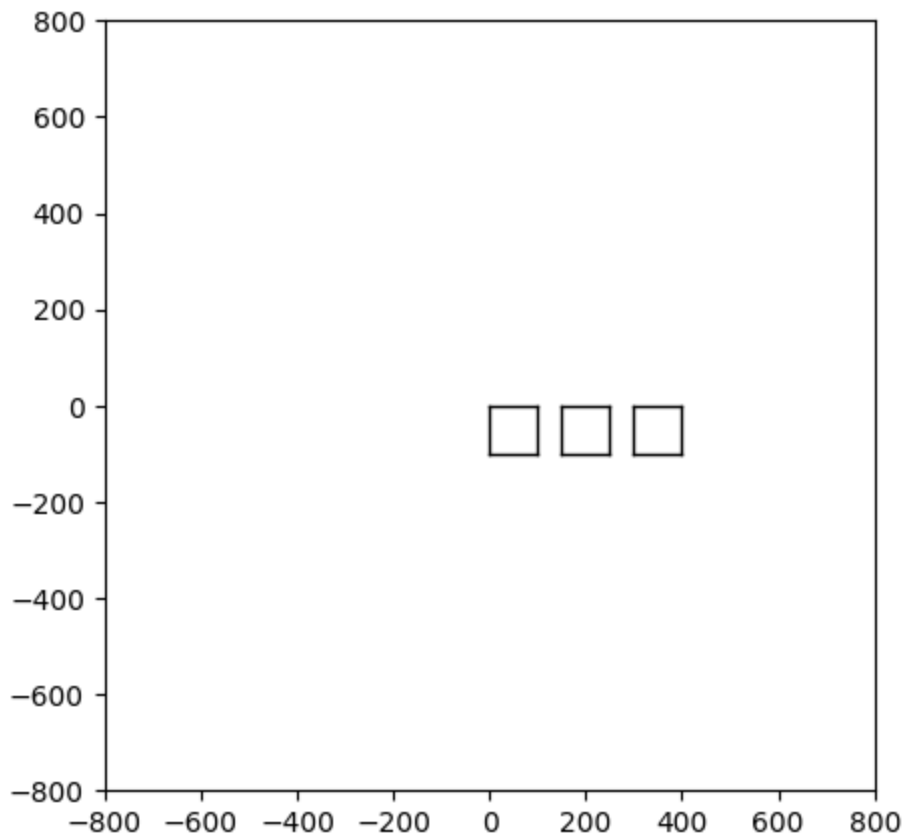
            if pen == "down":
                pendown()
            else:
                penup()

            original_forward(distance)

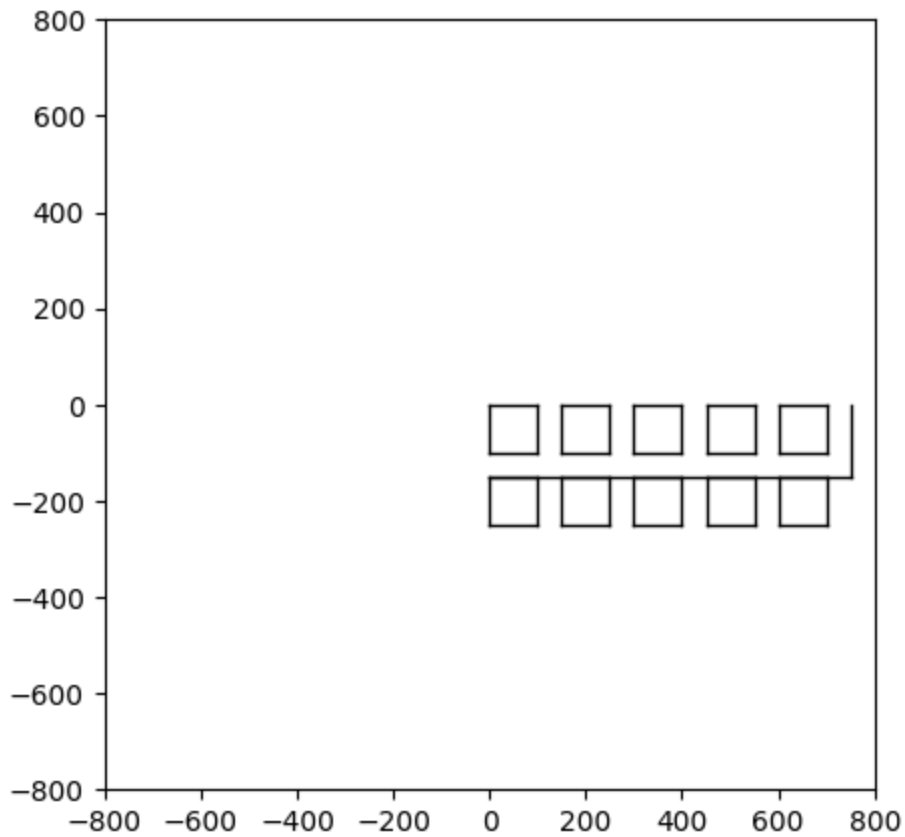
            if was_up:
                penup()
```

```
else:  
    pendown()
```

```
In [9]: reset()  
square(100)  
forward(150,"up")  
square(100)  
forward(150,"up")  
square(100)  
forward(150,"up")
```



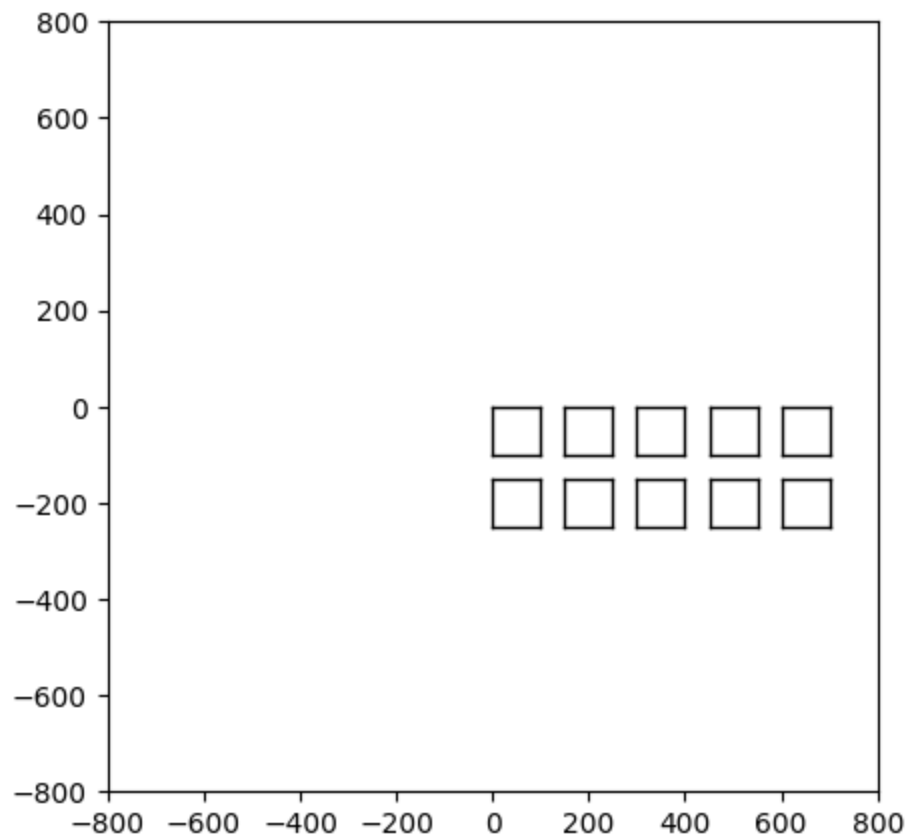
```
In [11]: reset()  
for i in range(5):  
    square(100)  
    forward(150,"up")  
  
right(90)  
forward(150)  
right(90)  
forward(5*150)  
right(180)  
  
for i in range(5):  
    square(100)  
    forward(150,"up")
```



```
In [12]: reset()
for i in range(5):
    square(100)
    forward(150, "up")

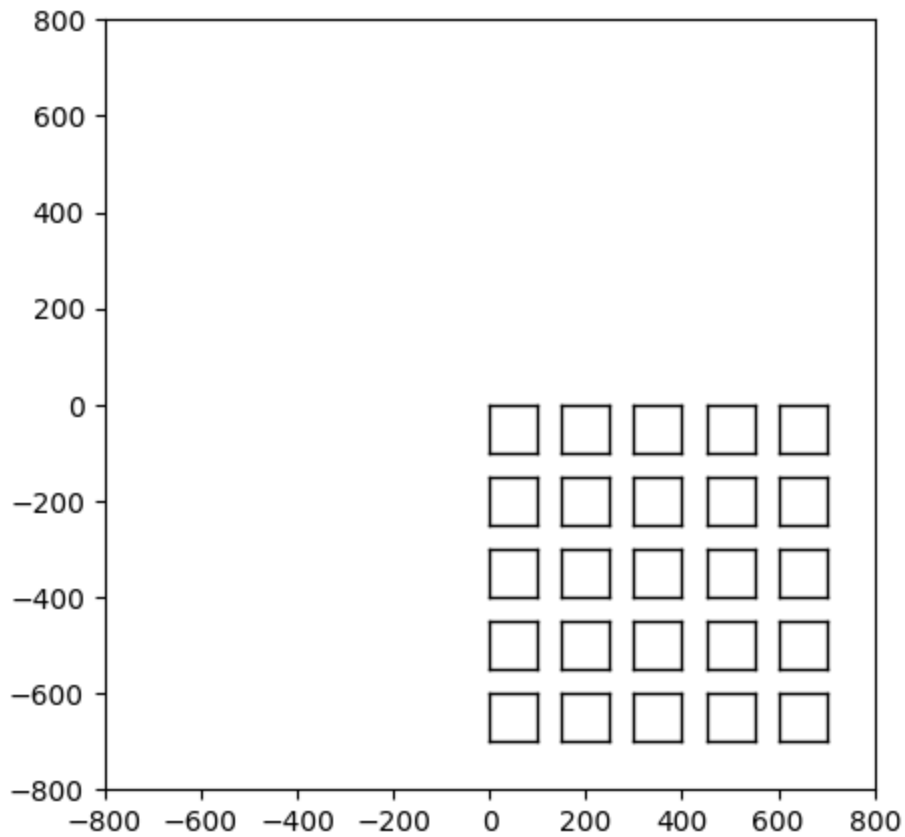
    right(90)
    forward(150, "up")
    right(90)
    forward(5*150, "up")
    right(180)

for i in range(5):
    square(100)
    forward(150, "up")
```



```
In [14]: reset()
for j in range(5):
    for i in range(5):
        square(100)
        forward(150, "up")

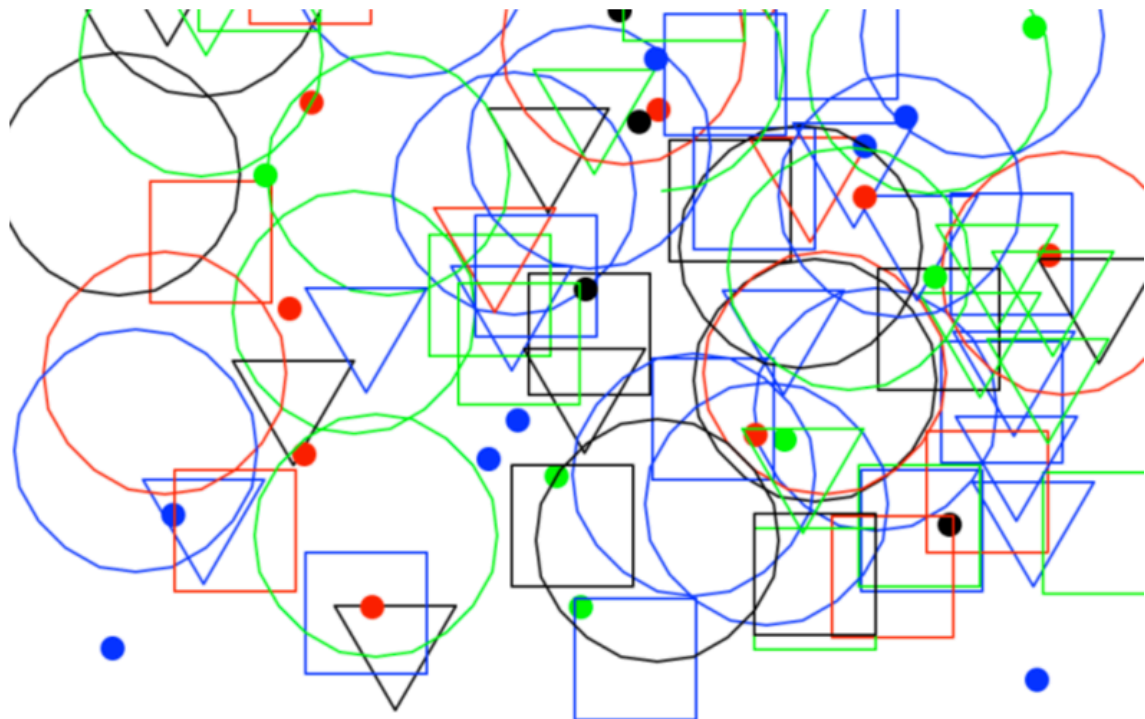
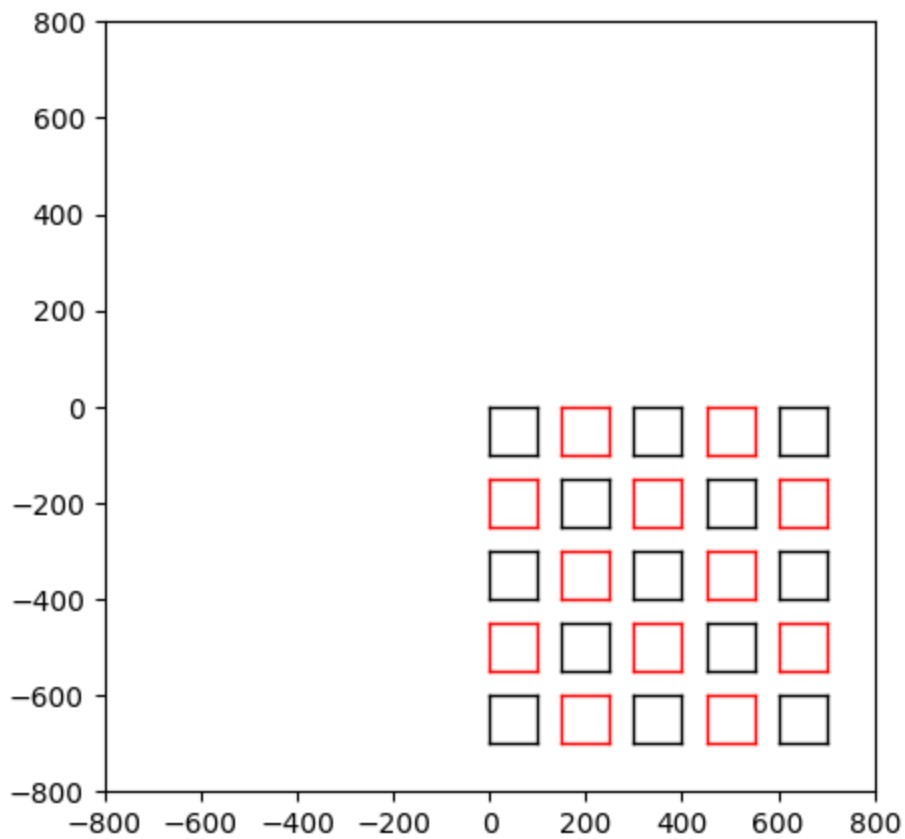
    right(90)
    forward(150, "up")
    right(90)
    forward(5*150, "up")
    right(180)
```



```
In [15]: reset()
count=0
for j in range(5):
    for i in range(5):
        if count%2==0:
            pencolor("black")
        else:
            pencolor("red")
        count=count+1

        square(100)
        forward(150,"up")

    right(90)
    forward(150,"up")
    right(90)
    forward(5*150,"up")
    right(180)
```



```
In [16]: from random import randint, choice
```

```
In [22]: reset()
```

```
for i in range(100):
    x,y=randint(-200,200),randint(-200,200)
```

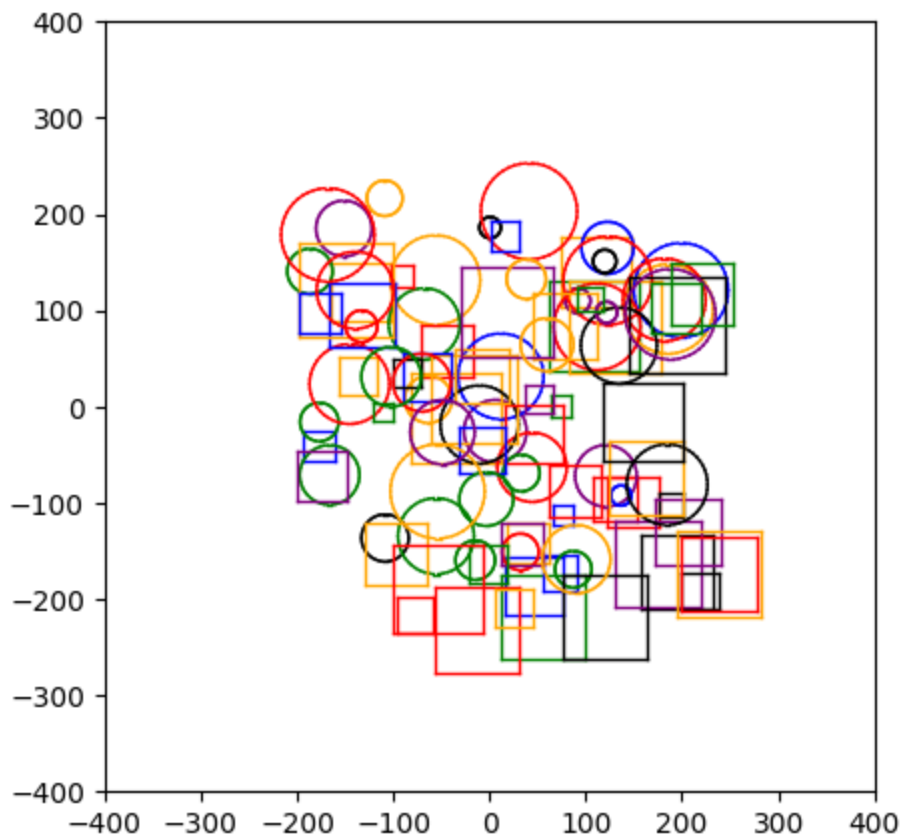
```

size=randint(20,100)
color=choice(["red","blue","green","black","orange","purple"])
shape=choice(["square","circle"])

penup()
goto(x,y)
pendown()

pencolor(color)
if shape=="square":
    square(size)
elif shape=="circle":
    circle(size)
else:
    raise ValueError("Unknown shape: "+shape)

```



In [ ]: