Can you write this in Python?

```c
#include <stdio.h>
int main(void)
{
    int count;
    for (count = 1; count <= 500; count++)
        printf("I will not throw paper airplanes in class.");
    return 0;
}
```
Modify \texttt{draw\_car} to allow the car to be drawn at any size.

- Existing car: size 100. Second car: \((150, 10)\), size 50.

Once you have this working, use a \texttt{for} loop with your function to draw a line of cars, like the picture at right.

- Start at \((10, 130)\), each size 40, separated by 50px.
Animation with `sleep`

- **DrawingPanel's sleep function** pauses your program for a given number of milliseconds.

- **You can use `sleep` to create simple animations.**
  ```python
  panel = DrawingPanel(250, 200)
  for i in range(1, NUM_CIRCLES + 1):
      panel.draw_oval(15 * i, 15 * i, 30, 30)
      panel.sleep(500)
  ```

- **Try adding `sleep` commands to loops in past exercises in this chapter and watch the panel draw itself piece by piece.**
def main():
    panel = DrawingPanel(260, 100, background="light gray")
    draw_car(panel, 10, 30, 100)
    draw_car(panel, 150, 10, 50)
    for i in range(0, 5):
        draw_car(panel, 10 + i * 50, 130, 40);

def draw_car(p, x, y, size):
    p.fill_rect(x, y, size, size / 2, "black")
    p.fill_oval(x + size / 10, y + size / 5 * 2, size / 5, size / 5, "red")
    p.fill_oval(x + size / 10 * 7, y + size / 5 * 2, size / 5, size / 5, "red")
    p.fill_rect(x + size / 10 * 7, y + size / 10, size / 10 * 3, size / 5, "cyan")
How to add parameters

• The panel must always be a parameter to a function that draws

• Add in position \((x, y)\) parameters
  • These change \(x\) and \(y\) but not width and height of figures

• Add \texttt{size} parameter
  • This changes width and height and sometimes \(x\) and \(y\)
  • Think of all sizes and placements as percentages of the size
    • \texttt{size (width) was 100, wheel was 70 from left, that is 70\% from the left so, size / 10 * 7}
Nested Loops

• What does the following code output?

def main():
    for i in range(1, 10):
        for j in range(1, 10):
            print(j * i, end="\t")
        print()

main()
Outputting a Grid

for i in range(n, m):
    for j in range(n, m):
        ...

controls the number of rows
creates a single row
Output the following figures

• Left grid:
  • $x = 100$
  • $y = 100$
  • circle size = 20
  • number of circles = 5

• Right grid:
  • $x = 300$
  • $y = 300$
  • circle size = 40
  • number of circles = 2