

Nested For Loops Practice (cont.)

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
    for(int j = 0; j < numSquares; j++) {  
        fill(i*255/numSquares,j*255/numSquares,0);  
        rect(j*width/numSquares, i*height/numSquares,  
            width/numSquares, height/numSquares);  
    }  
}
```

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 0`

`numSquares = 10`

`j = 0`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 0$

$\text{numSquares} = 10$

$j = 1$

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 0$

$\text{numSquares} = 10$

$j = 2$

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 0`

`numSquares = 10`

`j = 3`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 0$

$\text{numSquares} = 10$

$j = 4$

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 0`

`numSquares = 10`

`j = 5`


```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 0`

`numSquares = 10`

`j = 6`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 0$

$j = 7$

$\text{numSquares} = 10$

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 0$

$\text{numSquares} = 10$

$j = 8$

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 0`

`numSquares = 10`

`j = 9`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
        width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 0`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 1`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 2`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 3`

`numSquares = 10`


```
size(500,500);
noStroke();
int squareSize = 50;
int numSquares = width/squareSize;
for(int i = 0; i < numSquares; i++) {
  for(int j = 0; j < numSquares; j++) {
    fill(i*255/numSquares,j*255/numSquares,0);
    rect(j*width/numSquares, i*height/numSquares,
width/numSquares, height/numSquares);
  }
}
```

`i = 1`

`j = 4`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 5`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 6`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 7`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`j = 8`

`numSquares = 10`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

`i = 1`

`numSquares = 10`

`j = 9`

```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

$i = 2$

$\text{numSquares} = 10$

$j = 0$




```
size(500,500);  
noStroke();  
int squareSize = 50;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
  for(int j = 0; j < numSquares; j++) {  
    fill(i*255/numSquares,j*255/numSquares,0);  
    rect(j*width/numSquares, i*height/numSquares,  
width/numSquares, height/numSquares);  
  }  
}
```

i = 9

numSquares = 10

j = 9

Create a grid of squares, each randomly colored, using a for loop within a for loop.

```
void setup() { size(400,400); }
void draw() {
  //background(255);
  for(int i = 0;i < 40;i++){
    for(int j = 0; j < 40;j++){
      int r = (int)random(255);
      int g = (int)random(255);
      int b = (int)random(255);
      fill(r,g,b);
      rect(j*10, i*10, 10,10);
    }
  }
}
```

How many rectangles are drawn using the following code snippet? In other words, how many times does the nested for loop iterate?

- A. 25
- B. 40
- C. (20 * 20 =) 400
- D. (25 * 25 =) 625
- E. (40 * 40 =) 1600

```
size(500,500);  
noStroke();  
int squareSize = 20;  
int numSquares = width/squareSize;  
for(int i = 0; i < numSquares; i++) {  
    for(int j = 0; j < numSquares; j++) {  
        fill(i*255/numSquares,j*255/numSquares,0);  
        rect(j*width/numSquares, i*height/numSquares,  
            width/numSquares, height/numSquares);  
    }  
}
```

```

int squareSize = 20, numSquares;
void setup() {
    size(500,500);
    numSquares = width/squareSize;
    noStroke();
}
void draw() {
    for(int i = 0; _____; i++) {
        for(int j = 0; _____; j++) {
            fill(i*255/numSquares, j*255/numSquares, 0);
            rect(j*width/numSquares, i*height/numSquares,
                width/numSquares, height/numSquares);
        }
    }
}

```

Which choice makes the filled area more or less follow the mouse?

- A. $i < mouseY$ $j < mouseX$
- B. $i < mouseX$ $j < mouseY$
- C. $i < mouseY/squareSize$ $j < mouseX/squareSize$
- D. $i < mouseX/squareSize$ $j < mouseY/squareSize$