Nested For loops Practice

Ch. 6 Continuation

Simple `for` Statements (recap)

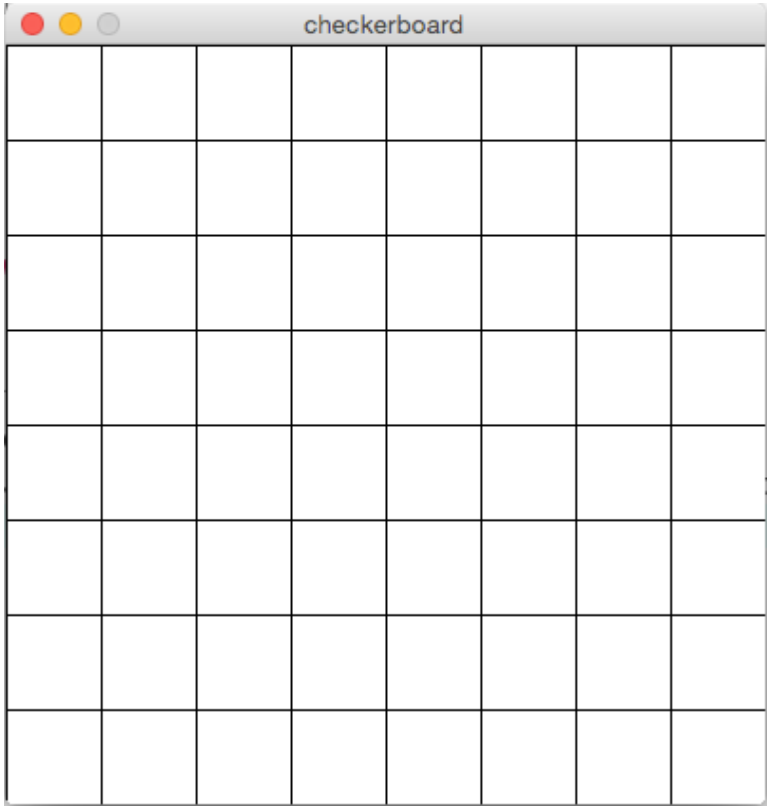
```
// print numbers 0 to 9
for(int value = 0; value <= 9; value = value+1) {
   println(value);
}

// draw vertical lines – assume xDist is already declared
for (int xPos = 0; xPos < width; xPos = xPos+xDist) {
   line(xPos,0, xPos, height);
}
```

Simple `for` Statements (recap)

```
size(500, 500);
for(int i = 1; i <= 100; i++) {
   ellipse(random(width), random(height),
   random(width/10), random(height/10));
}
```

Nested `for` loops




```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
  size(400,400);
  numRows = 8;
  squareWidth = width/numRows;
}
void draw(){
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      rect(j*squareWidth,i*squareWidth,squareWidth,squareWidth);
    }
  }
}//How many squares are being drawn in this program?
```

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

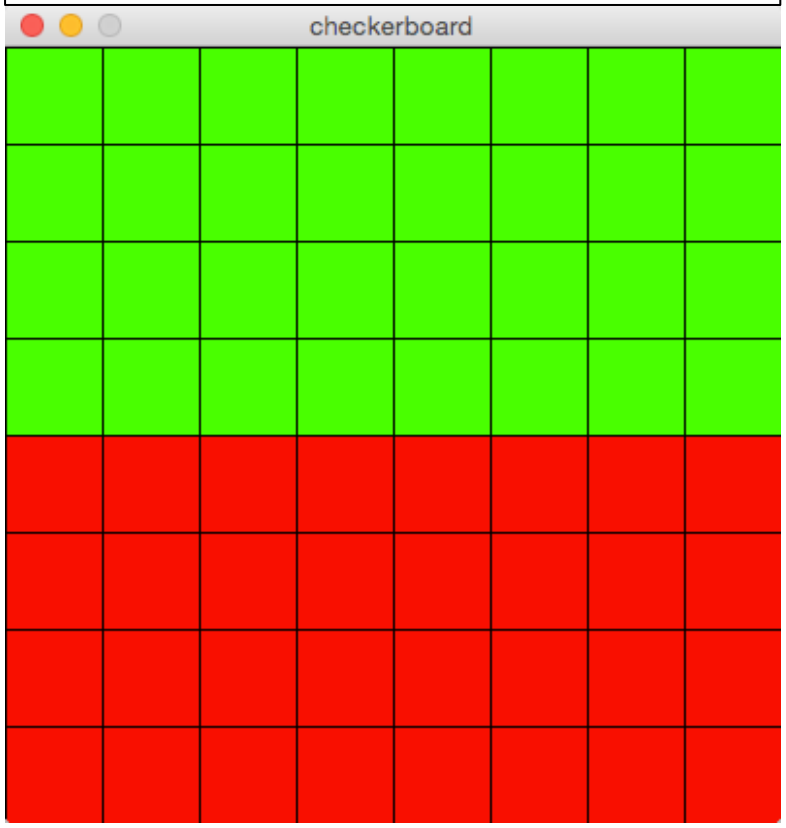
C. (20 * 20 =) 400

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);
   }
}
```

Add the missing code to this program so that it produces the image below.

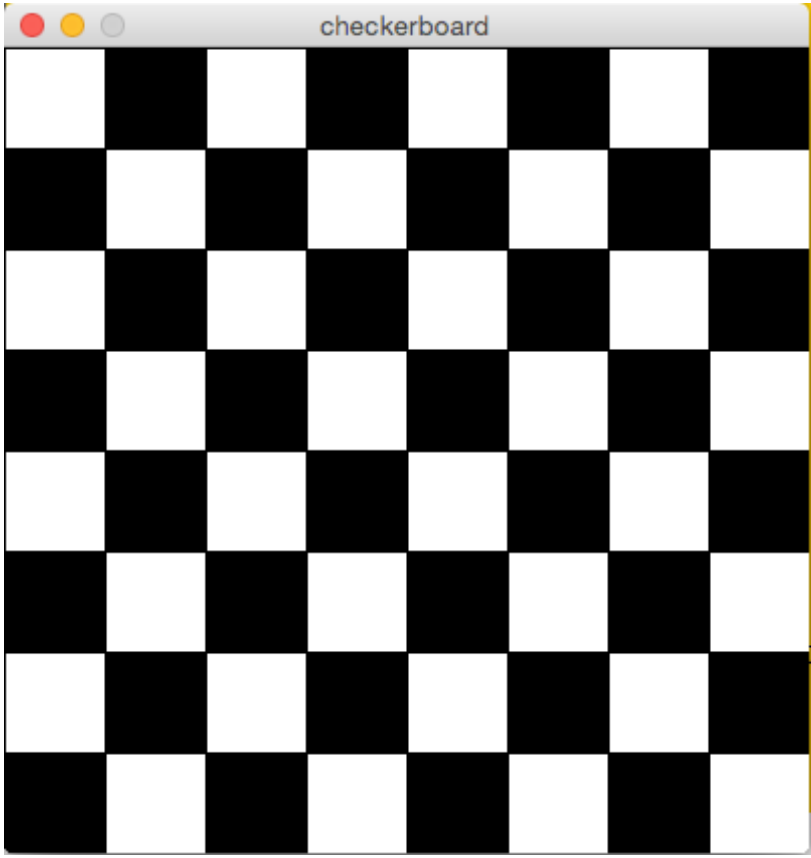



```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
  size(400,400);
  numRows = 8;
  squareWidth = width/numRows;
}
void draw(){
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      //Begin Missing Code



      //End Missing Code
      rect(j*squareWidth,i*squareWidth,squareWidth,squareWidth);
    }
  }
}
```

Use two for loops (nested) to draw a checkerboard as shown below.



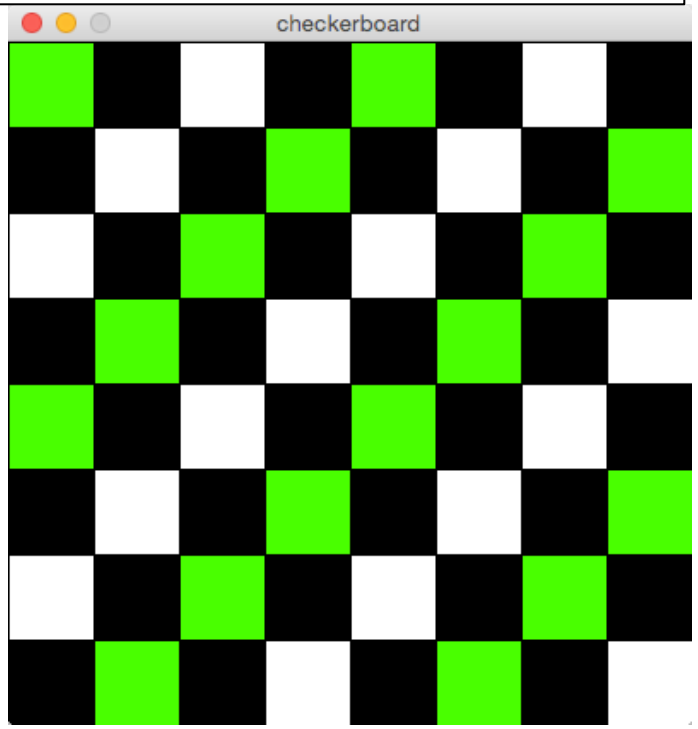

```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
    size(400,400);
    numRows = 8;
    sqWidth = width/numRows;
}
void draw(){
  background(0);
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      if(_____________){
        rect(j*sqWidth, i*sqWidth, sqWidth, sqWidth);
      }
    }
  }
}
```

```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
   size(400,400);
   numRows = 8;
   squareWidth = width/numRows;
}
void draw(){
  background(0);
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      if((i+j)%2 == 0){
        rect(j*squareWidth, i*squareWidth, squareWidth,
         squareWidth);
      }
    }
  }
}
```

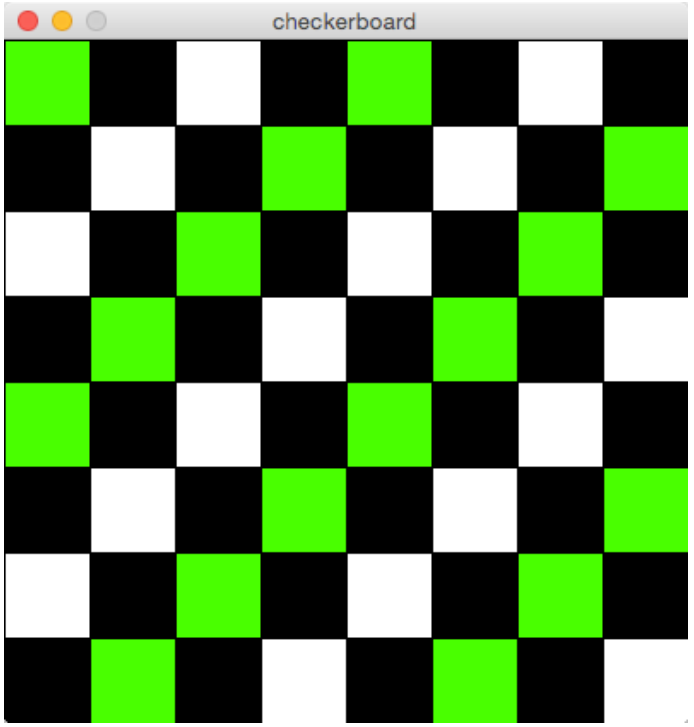
```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
  size(400,400);
  numRows = 8;
  squareWidth = width/numRows;
}
void draw(){
  background(0);
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      if((i+j)%2 == 0){
        //Begin Missing Code


        //End Missing Code
        rect(j*squareWidth,i*squareWidth,squareWidth,squareWidth);
      }
      fill(255);
    }
  }
}
```

Add the missing code to this program such that the white squares from the previous program alternate between white and green, as shown here.



```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
  size(400,400);
  numRows = 8;
  squareWidth = width/numRows;
}
void draw(){
  background(0);
  for(int i = 0;i < numRows;i++){
    for(int j = 0;j < numRows;j++){
      if((i+j)%2 == 0){
        if((i+j)%4 == 0){
          fill(0,255,0);
        }
        rect(j*squareWidth,i*squareWidth,squareWidth,squareWidth);
      }
      fill(255);
    }
  }
}
```



```
//Program draws a grid of squares with fading colors
int squareSize = 20, numSquares;
void setup() {
  size(500,500);
  numSquares = width/squareSize;
  noStroke();
}
void draw() {
  background(255);
  for(int i = 0; i < mouseY/squareSize; i++) {
    for(int j = 0; j < mouseX/squareSize; j++) {
      fill(i*255/numSquares,j*255/numSquares,0);
      rect(j*width/numSquares, i*height/numSquares,
      width/numSquares, height/numSquares);
    }
  }
}
```

```
//Draw a checkerboard
int squareWidth;
int numRows;
void setup(){
  size(400,400);
  numRows = 8;
  squareWidth = width/numRows;
  noFill();
}
void draw(){
  for(int i = 0;i < numRows;i++){
    for(int j = i;j < numRows;j++){
      rect(j*squareWidth,i*squareWidth,
           squareWidth,squareWidth);
    }
  }
}
```

How many squares are drawn in this program?

```
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`