



ESTRUTURAS DE REPETIÇÃO

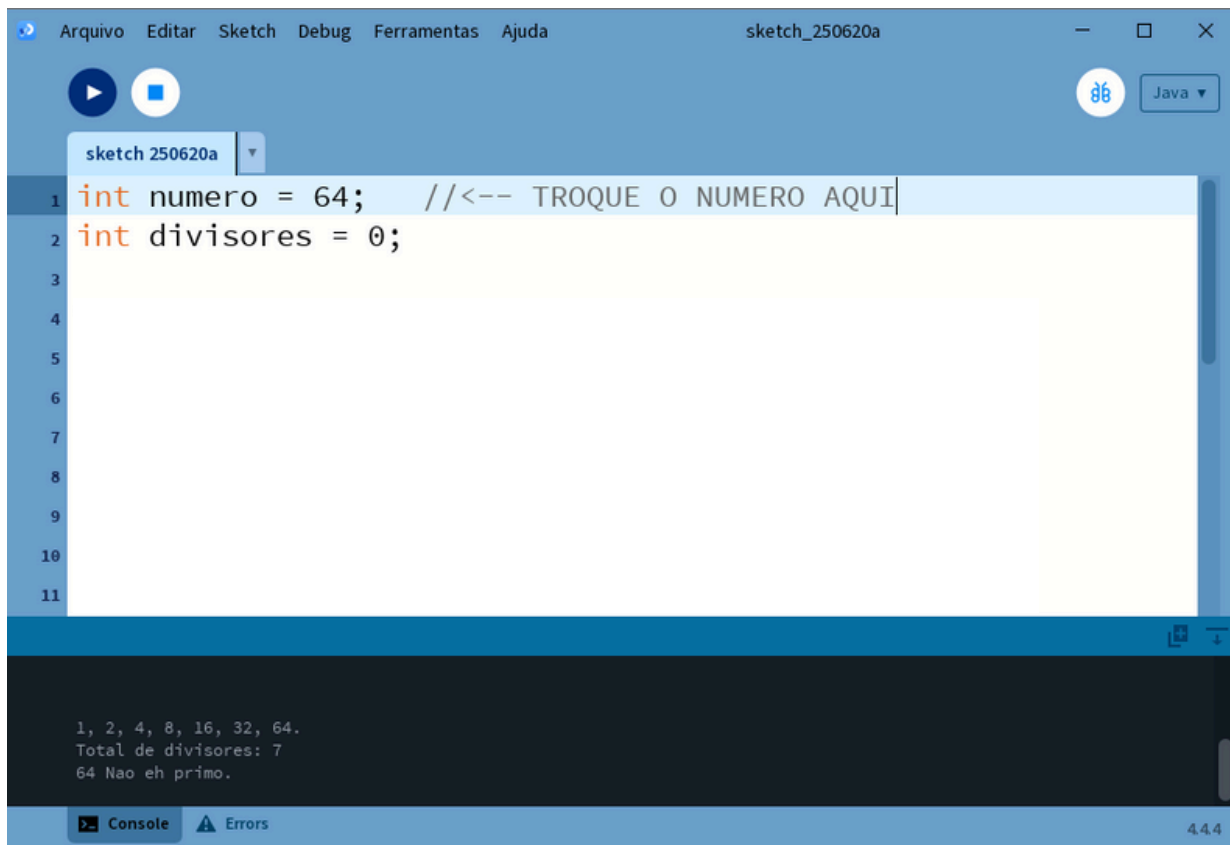
LAÇO FOR

1) Vamos exercitar a lógica usando o laço de repetição ‘for’. Para isso, resolva os seguintes itens:

- a) Imprima os números de 1 a 20 no console com *println()*.
- b) Em um intervalo de 1 a 20, imprima no console os números pares com *println()*.
- c) Em um intervalo de 1 a 20, imprima no console os números ímpares com *println()*.
- d) Em um intervalo de 1 a 100, imprima no console os números divisíveis por 5.
- e) Imprima no console os quadrados dos números de 1 a 10 (ex: 1, 4, 9...).
- f) Imprima apenas os números entre 30 e 60 que são divisíveis por 3.
- g) Imprima os números de 50 até 1 em ordem decrescente.
- h) Imprima os números de 0 a 128 pulando de 8 em 8.
- i) Some todos os números de 1 a n, onde n é um número inteiro aleatório e imprima o resultado final. (ex: int n = 10; SAÍDA: 55).
- j) Usando um contador, conte quantos números entre 1 e 100 são divisíveis por 7, e imprima apenas a quantidade final.

k) Escreva um programa em Processing que:

1. Utilize uma variável arbitrária “n” do tipo inteiro de sua preferência (ex: n = 64)
2. Imprima todos os divisores positivos desse número;
3. Conte a quantidade total de divisores;
4. Ao final, informe se o número é primo ou não primo (um número primo possui exatamente dois divisores: 1 e ele mesmo).



The screenshot shows the Processing IDE interface. The menu bar includes 'Arquivo', 'Editar', 'Sketch', 'Debug', 'Ferramentas', and 'Ajuda'. The title bar says 'sketch_250620a'. The toolbar has a play button, a stop button, and a language dropdown set to 'Java'. The code editor shows the following code:

```
1 int numero = 64; //<-- TROQUE O NUMERO AQUI
2 int divisores = 0;
3
4
5
6
7
8
9
10
11
```

The console output at the bottom shows:

```
1, 2, 4, 8, 16, 32, 64.
Total de divisores: 7
64 Nao eh primo.
```

Exemplo de saídas:

Exemplo 01:

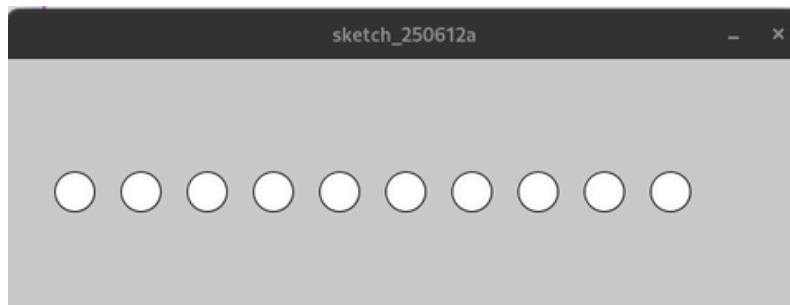
1, 2, 4, 8, 16, 32, 64.	// todos os divisores positivos
Total de divisores: 7	// número total dos divisores
64 Não é primo.	// o número é primo ou não

Exemplo 02:

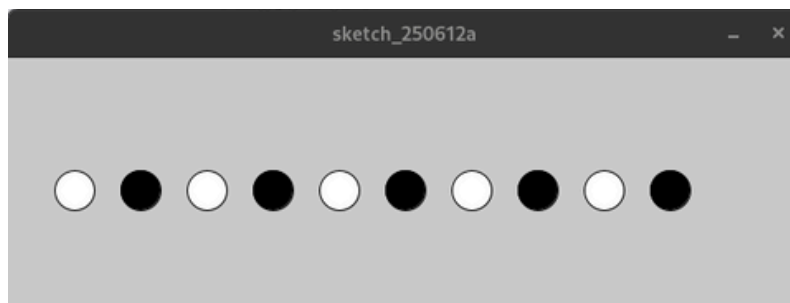
1, 67.	// todos os divisores positivos
Total de divisores: 2	// número total de divisores
67 eh um primo.	// o número é primo ou não

2) Resolva o seguinte problema utilizando o laço de repetição “for”

a) Desenhe 10 círculos espaçados horizontalmente.

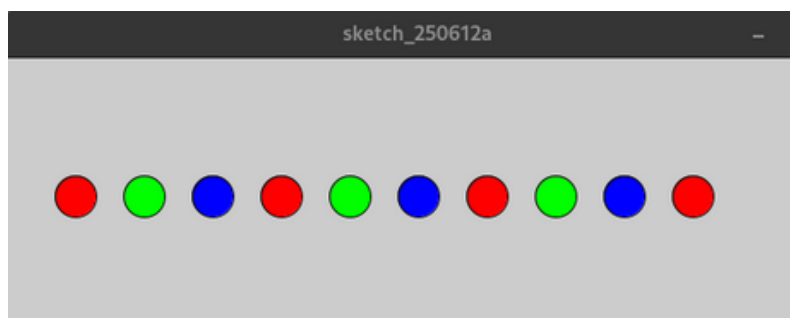


b) Agora faça uma estrutura condicional “if” para alternar as cores das elipses entre preto e branco.



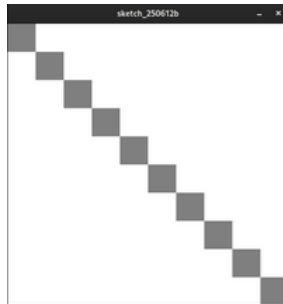
Dica: Use o operador módulo %, para auxiliar a verificar se um número é ímpar ou par.
Algoritmo ímpar ou par: [≤https://dicasdeprogramacao.com.br/exercicio-algoritmo-par-ou-impar/>](https://dicasdeprogramacao.com.br/exercicio-algoritmo-par-ou-impar/)

c) Reescreva o código anterior para alternar as cores das elipses entre **vermelho**, **verde** e **azul**.

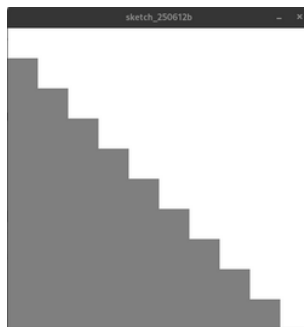


2) Utilizando o laço de repetição “for” e as primitivas geométricas do Processing, resolva os seguintes itens:

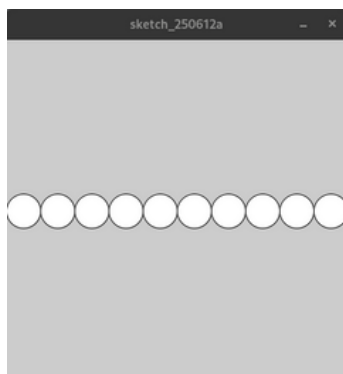
a) Usando a primitiva geométrica *rect()*. Faça uma "escada" com quadrados descendo na diagonal.



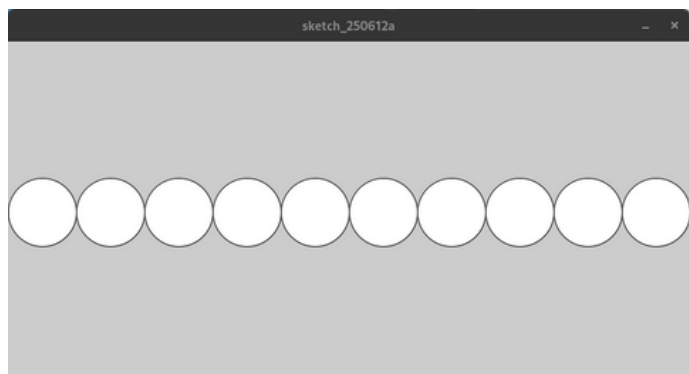
b) Agora ajuste os parâmetros da função *rect()* para alongar os “degraus” da escada



3) Desenhe uma série de círculos igualmente distribuídos horizontalmente no meio da tela. O código está estruturado de forma que funciona corretamente mesmo que a largura da janela (width) seja alterada, mantendo o espaçamento proporcional entre os círculos. Exemplo:



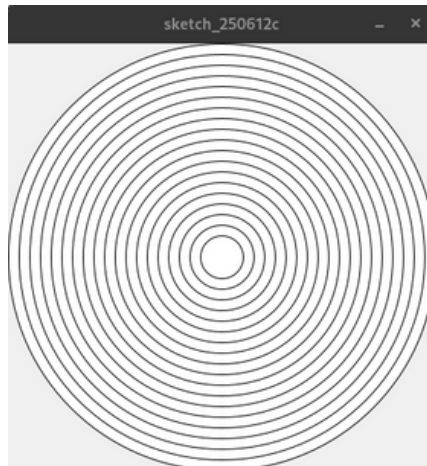
size(400, 400)



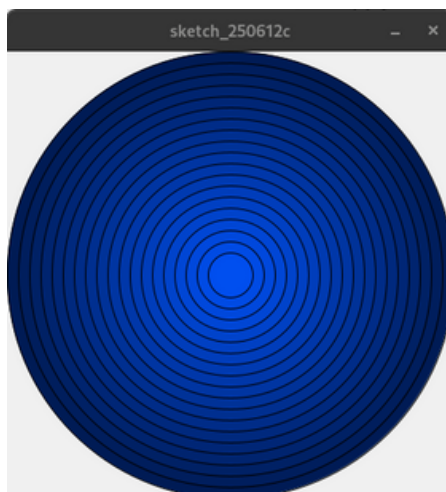
size(800, 400)

4) Utilizando o laço de repetição 'for' e a primitiva geométrica *ellipse()* resolva os seguintes itens:

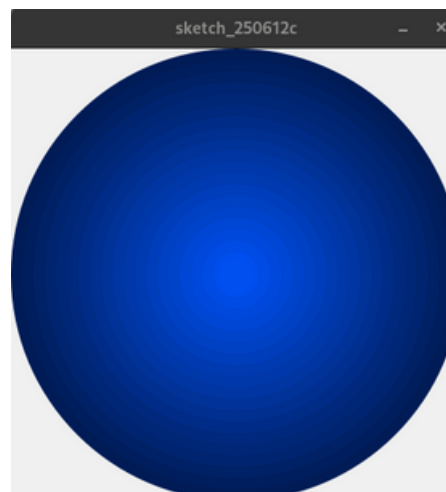
a) Desenhe vários círculos concêntricos **do maior para o menor**, com o centro fixo no meio da tela.



b) Utilizando o modo de cor HSB, pinte as elipses com uma matiz de sua preferência. Em seguida, altere o parâmetro de brilho da função *fill()* para gerar um degradê monocromático, onde as cores mais claras se concentram no centro.



com *stroke(0)*

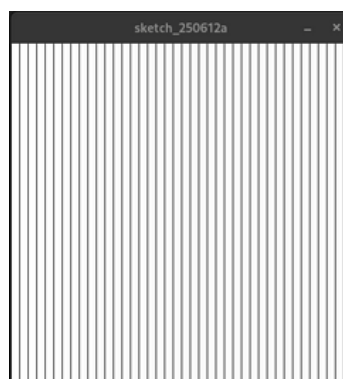


com *noStroke()*

5) Crie um código em Processing que preencha toda a tela com um retângulo, cuja cor no modo HSB varia continuamente da esquerda para a direita, como um arco-íris horizontal.



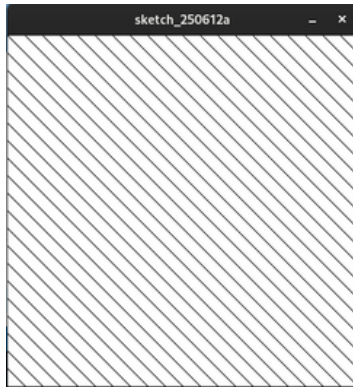
6) Crie um código em Processing que desenhe linhas verticais igualmente espaçadas, preenchendo toda a tela. As linhas devem ir do topo até a base da tela.



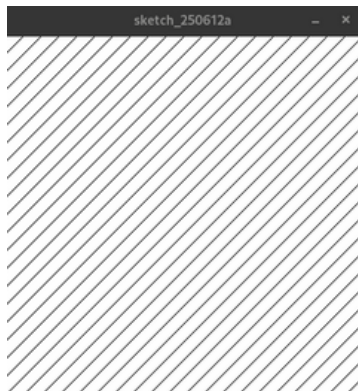
7) Usando um laço for, desenhe linhas horizontais igualmente espaçadas do lado esquerdo até o lado direito da tela.



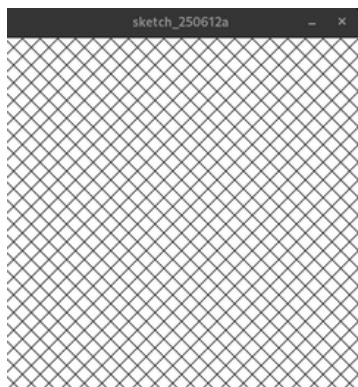
8) Crie um padrão de linhas diagonais inclinadas para baixo e para a direita (↘) que cubra toda a tela.



9) Crie um padrão de linhas diagonais inclinadas para baixo e para a esquerda (↙) preenchendo a tela.



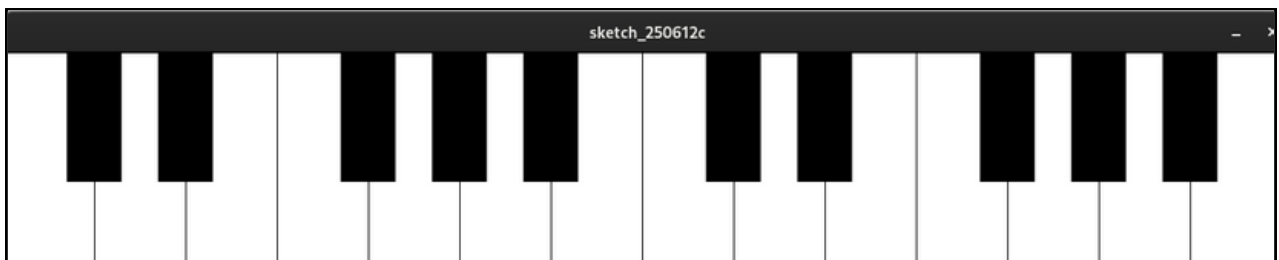
10) Desenhe duas séries de linhas diagonais cruzadas, formando um padrão em forma de X. As linhas devem se sobrepor e cobrir toda a área da tela.



🔥 **DESAFIO:** Crie um código em Processing que desenhe um teclado de piano com 14 teclas brancas verticais, igualmente espaçadas ao longo da largura da tela. Sobre essas teclas, desenhe teclas pretas mais estreitas e mais baixas, intercaladas de forma a lembrar o padrão de um teclado real: duas pretas, espaço, três pretas, espaço... Utilize apenas laços for e coordenadas cartesianas (sem funções trigonométricas).

Obs:

- As teclas pretas não aparecem sobre todas as brancas.
- As teclas pretas devem estar entre duas brancas.



LAÇO FOR ANINHADO

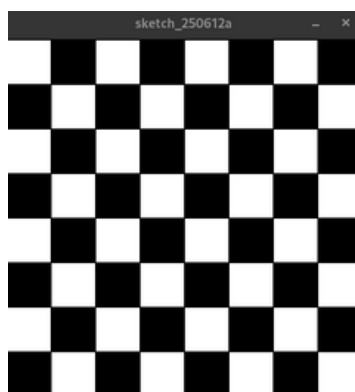
11) Em um intervalo de 1 a 50, imprima no console os números primos utilizando a função `println()` e a estrutura de repetição `for`.

Dica: Um número primo é aquele que só é divisível por 1 e por ele mesmo.

12) Desenhe uma grade 5x5 de quadrados na tela usando `rect()`. Todos devem ter o mesmo tamanho e espaçamento regular.

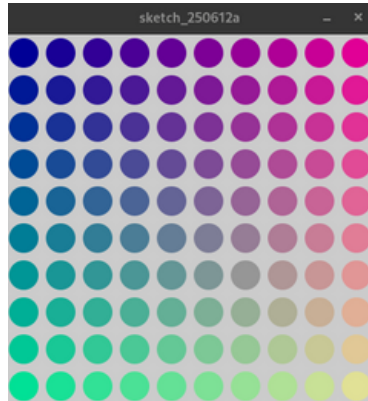


13) Use `for` aninhado para desenhar um tabuleiro de xadrez 8x8. As cores devem se alternar entre branco e preto em cada casa.



Dica: Use o operador módulo `%`, para auxiliar a verificar se um número é ímpar ou par.
Algoritmo ímpar ou par: <https://dicasdeprogramacao.com.br/exercicio-algoritmo-par-ou-impar/>

14) Crie uma grade 10x10 de círculos usando `ellipse()`, alterando o parâmetro `fill()` para variar a cor de preenchimento em cada linha ou coluna.

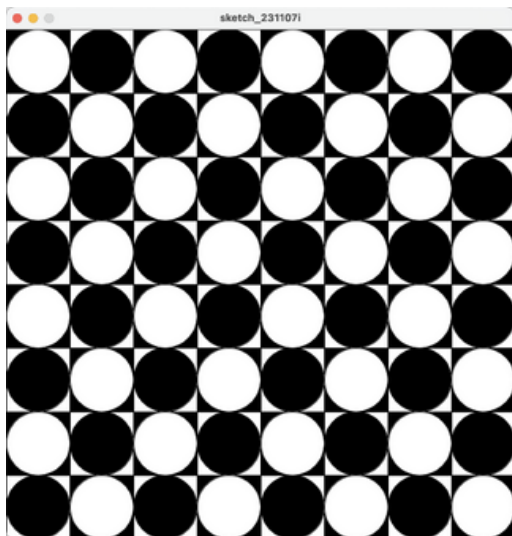


15) Os murais abaixo foram criados por alunos de semestres anteriores usando o Processing com laços `for` aninhados. Observe cada imagem com atenção e tente reproduzir os murais em cada item.

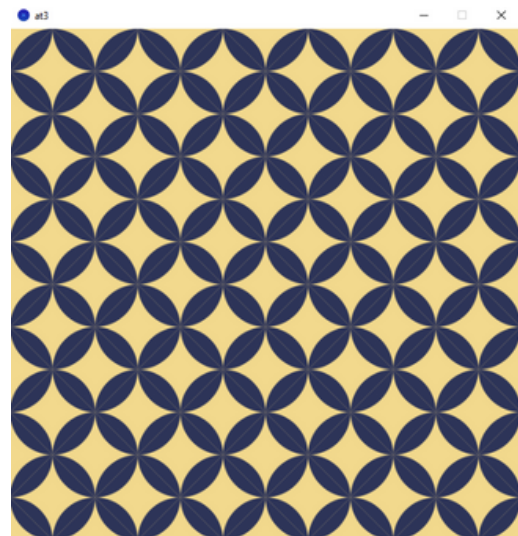
Use seus conhecimentos sobre laços de repetição duplos (`for` aninhados), combinações de primitivas gráficas (`rect()`, `ellipse()`, `fill()`, etc.) e organização de espaço na tela para responder corretamente.

Dica: Use o operador módulo `%` para auxiliar a alternar os padrões das formas

(a)



(b)

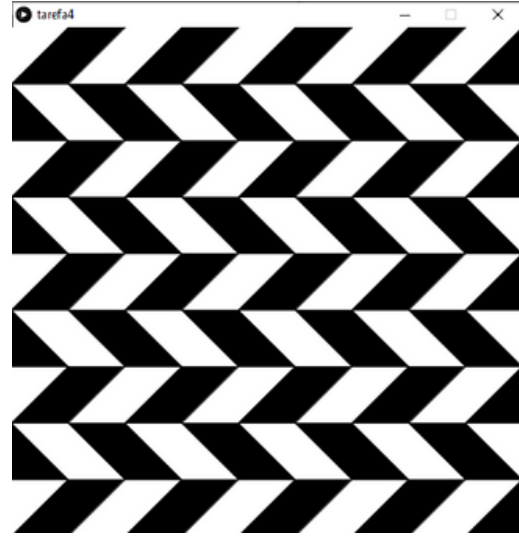


Dica: Utilize o parâmetro `CHORD` da função `arc()`

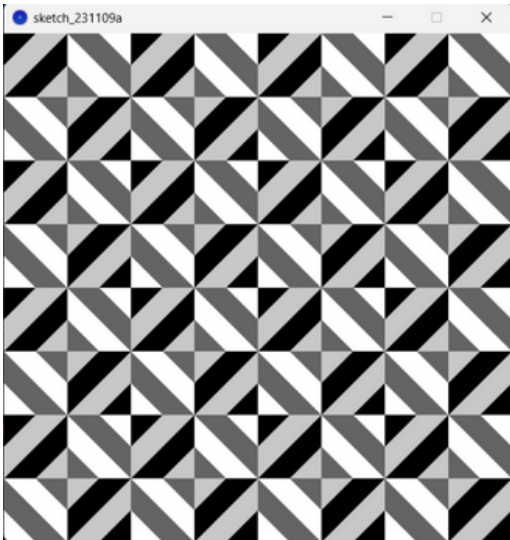
(c)



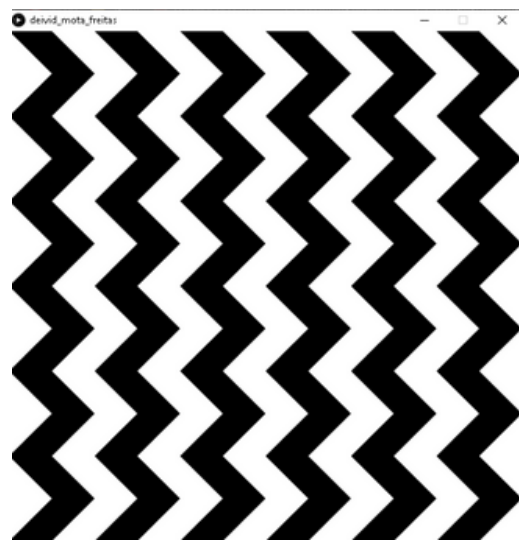
(d)



(e)



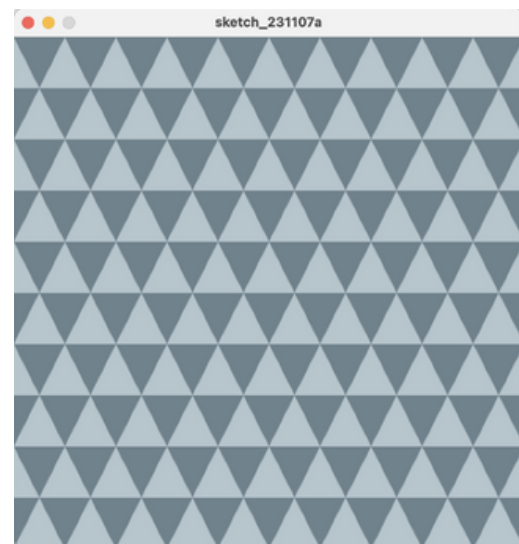
(f)



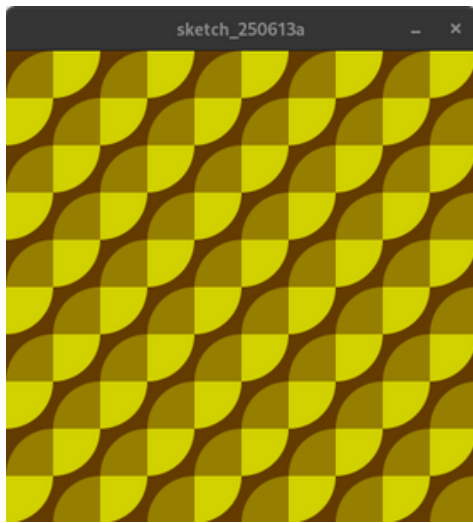
(g)



(h)



(c)



(i)

