

Lista de Exercícios - Estruturas de Repetição - Parte 2

1. Faça um algoritmo que mostre os números de -10 a 10, em ordem crescente.
2. Faça um algoritmo que mostre os números de -10 a 10, em ordem decrescente.
3. Faça um algoritmo que calcule o somatório de 1 a 4.
4. Faça um algoritmo que calcule o somatório entre dois números obtidos.
Exemplo: dados os números 12 e 7, calcular o somatório de 7 até 12 e mostrar o resultado (que é 57).
OBS: Perceba que, no exemplo, foi necessário ajustar a ordem dos números obtidos.
5. Faça um algoritmo que mostre os números entre 100 e 200 que são pares e também divisíveis por 5.
6. Faça um algoritmo que mostre os números entre 100 e 200 que são pares e também divisíveis por 5. Mostre, também, a quantidade de valores que atenderam a esses requisitos.
7. Faça um algoritmo capaz de produzir a seguinte saída:



8. Faça um algoritmo capaz de produzir a seguinte saída:



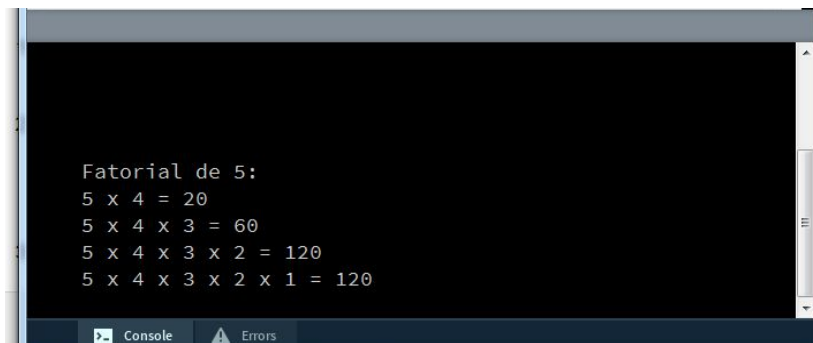
9. Faça um algoritmo capaz de produzir a seguinte saída:



10. Modifique os algoritmos das questões 7, 8 e 9 para que ele seja capaz de criar um desenho conforme um valor fornecido. Por exemplo, na questão 7, se for fornecido o valor 6, o algoritmo “desenhará” um “quadrado de lado 6”; na questão 9, seria “desenhado” um “triângulo com altura 6”. A figura abaixo mostra um exemplo de saída da questão 7 para o número 6 fornecido.



11. Faça um programa que calcule o fatorial de um número n, exibindo cada um de seus resultados parciais. Um exemplo de saída seria:



12. Dado um número n informado no programa, calcule a soma definida pela sequência para a seguinte expressão:

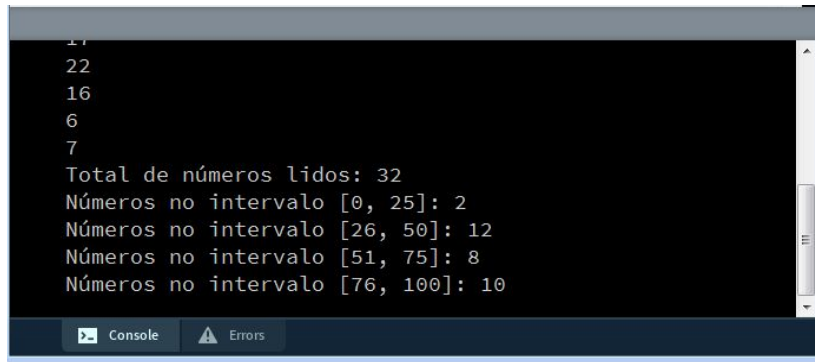
$$S = 1 + 1/2 + 1/3 + 1/4 + 1/5 + \dots + 1/n$$

13. Chico tem 1,50 metro e cresce 2 centímetros por ano, enquanto Zé tem 1,10 metro e cresce 3 centímetros por ano. Faça um programa que calcule e imprima quantos anos serão necessários para que Zé seja maior que Chico.

14. Faça um programa que obtenha uma quantidade desconhecida de números e conte quantos deles estão nos seguintes intervalos: [0, 25], [26,50], [51,75] e [76,100]. A entrada de dados deve terminar quando for lido um número negativo.

OBS: No processing, a leitura dos dados pode ser feita por comandos de atribuição de valores aleatórios.

OBS2: Um exemplo de saída seria:



```
11
22
16
6
7
Total de números lidos: 32
Números no intervalo [0, 25]: 2
Números no intervalo [26, 50]: 12
Números no intervalo [51, 75]: 8
Números no intervalo [76, 100]: 10
```

15. Faça um programa capaz de registrar votos em uma determinada eleição. Assuma que temos os candidatos 1, 2, 3 e 4. O eleitor deverá entrar com um desses valores para registrar um voto válido. Caso o eleitor entre com o valor 0 (zero), o programa deverá encerrar e exibir um relatório com as seguintes informações:

- A quantidade total de eleitores;
- O resultado da eleição: quantos votos cada um dos candidatos obteve;
- Mostrar qual candidato foi o vencedor.

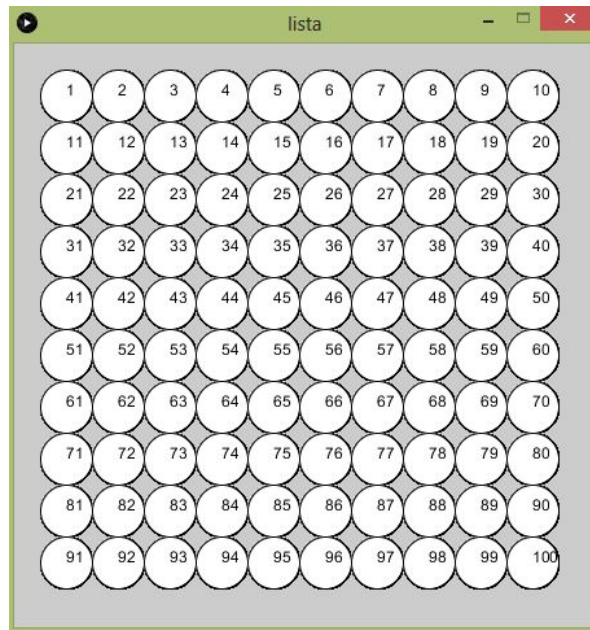
Dica: o voto de cada eleitor pode ser determinado por meio de geração de valores aleatórios.

16. Usando a função `line()` do Processing, escreva um programa que seja capaz de de desenhar 5 triângulos, cada um em uma cor de linha diferente.

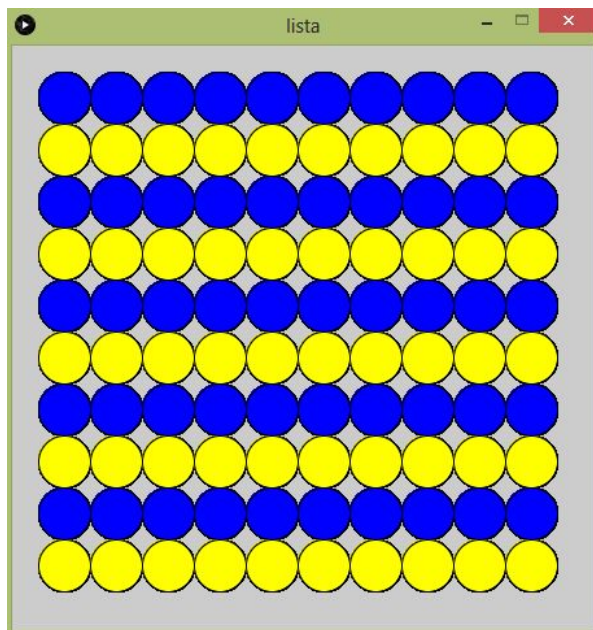
Dica: a geração dos vértices de cada triângulo pode ser feita de maneira aleatória. Lembrar que os vértices deverão estar dentro dos limites do tamanho da janela criada por seu programa.

Dica2: a geração das cores pode ser feita de maneira aleatória.

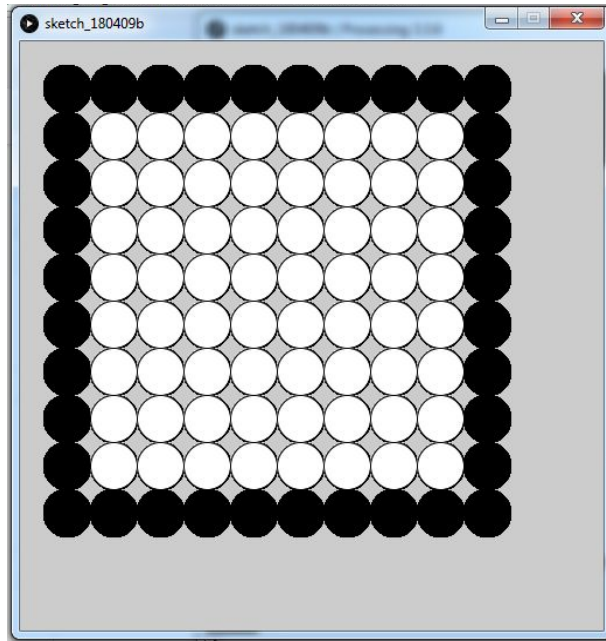
17. Faça um programa que desenhe uma matriz de círculos de tamanho 10 x 10 e exiba dentro de cada círculo o valor de 1 a 100.



18. Faça um programa que desenhe uma matriz de círculos de tamanho 10 x 10. Pinte de azul os círculos de linhas ímpares e de amarelo os círculos das linhas pares.



19. Usando estruturas de repetição e de decisão, faça um programa que replique a figura abaixo:



20. Faça um programa que desenhe 4 retângulos de cores diferentes (tanto da linha como do preenchimento).
21. Faça um programa que desenhe 10 pontos em posições aleatórias na tela do Processing. Cada ponto deverá estar com tamanho 8 e deverá ter uma cor diferente. **Dica:** pesquise a função `strokeWeight()` do Processing, para ajustar o tamanho dos pontos.
22. Faça um programa que:
- mude o tamanho da tela do Processing;
 - exiba um texto na tela dizendo: “pressione uma tecla”
 - sempre que o usuário pressionar uma tecla, um ponto com posição aleatória deverá aparecer na tela.
 - o programa encerrará quando o usuário pressionar a tecla ‘f’ (de fim). :-) Ao encerrar, o texto “Fim” será exibido na mesma tela onde os pontos foram desenhados.

OBS: Para reconhecer o pressionamento de uma tecla, seu código de desenho de pontos e reconhecimento de uso de teclado deverá ser escrito dentro da função `void draw()` do Processing.

OBS2: Note que o pressionamento de uma tecla pode ser interpretado pelo seu sistema operacional como sendo o pressionamento dessa tecla diversas vezes. Então, na execução do seu programa, é possível que mais de um ponto seja desenhado a cada pressionamento de tecla individual. No momento, não vamos nos preocupar com isso.

OBS3: É sugerida a pesquisa sobre a função `text` do Processing, para atender ao problema proposto.

Dica: (novidade) Para encerrar o programa, ou seja, sair da função `draw()` e encerrar o laço infinito que essa função executa, você pode utilizar o comando `noLoop()` ;

trilha sonora sugerida:

<https://www.youtube.com/watch?v=X0B1qHADa-E>

Bom trabalho!!