

Lógica de Programação

Aula 10 - Exercícios de Estruturas Aninhadas: enquanto e para



Apresentação da Aula

Em aulas anteriores, você aprendeu que, em alguns casos, é necessário utilizar o **aninhamento** das estruturas de repetição. Nesta aula, você irá exercitar a construção de algoritmos em que a solução pode ser construída com o **aninhamento** das estruturas de repetição **enquanto** e **para**.



Objetivos

Exercitar a utilização das estruturas de repetição **enquanto**.

Exercitar a utilização das estruturas de repetição **para**.



Atividade 01

Comece pelo Desafio! Vá na sequência de 6.

Crie um algoritmo que recebe uma quantidade indefinida de números inteiros aleatórios. O programa encerra quando uma sequência crescente de 6 números é informada. Ao final, o programa escreve a sequência dos 6 números, um abaixo do outro.

São exemplos de sequências válidas para encerrar o programa:

1, 2, 3, 4, 5, 6.

19, 20, 21, 22, 23, 24.

Não são sequências válidas para encerrar o programa:

1, 2, 4, 5, 6, 7.

21, 23, 25, 27, 28, 29.



Atividade 02

Imprimindo pontos

Utilizando o aninhamento de **para**, crie um programa que recebe um valor inteiro e desenha um retângulo de tamanho Z por Z, utilizando o caractere *.

Fique atento! Há apenas um espaço em branco entre os caracteres * de uma mesma linha.

Veja o exemplo para uma entrada igual a 2.

```
* *  
* *
```



Atividade 03

Construindo um retângulo de pontos

Utilizando o aninhamento de **enquanto**, crie um programa que recebe dois valores inteiros e desenha um retângulo de tamanho Z (linhas) por W (colunas), utilizando o caractere *****.

Fique atento! Há apenas um espaço em branco entre os caracteres ***** de uma mesma linha.

Veja o exemplo para uma entrada igual a 6 e 4.

```
* * * *  
* * * *  
* * * *  
* * * *  
* * * *  
* * * *
```



Atividade 04

Triângulo

Utilizando o aninhamento de **enquanto**, implemente um programa que recebe um número inteiro e escreve o seguinte triângulo. A altura da figura será igual ao número inteiro recebido pelo programa.

Fique atento! Há apenas um espaço em branco entre os caracteres ***** de uma mesma linha.

```
*  
* *  
* * *  
* * * *
```



Atividade 05

Primo ou composto!

Escreva um programa que recebe um número inteiro. Seu programa irá verificar, do 2 até o número informado, quais são os números primos e quais não são.

Caso seja primo, imprima como no exemplo abaixo:

```
11 primo
```

Se o número não for primo o programa mostra o número seguido da palavra composto e abaixo a lista de seus divisores, com exceção do 1 e dele mesmo, como no exemplo a abaixo:

```
10 composto
2
5
```

Considerando a entrada igual a 6, a saída esperada para o seu algoritmo é:

```
2 primo
3 primo
4 composto
2
5 primo
6 composto
2
3
```



Atividade 06

Triângulo 2

Utilizando o aninhamento de **para**, implemente um programa que recebe um número inteiro e escreve o seguinte triângulo. A altura da figura será igual ao número inteiro recebido pelo programa.

Fique atento! Há apenas um espaço em branco entre os caracteres * de uma mesma linha.

```
* * * * *
* * * *
* * *
* *
*
```



Atividade 07

Sequência interrompida

Faça um programa que recebe uma quantidade indefinida de números inteiros até que o 0 (zero) seja informado. Seu programa deverá procurar pela maior sequência numérica existente.

São exemplos de sequências:

3, 4, 5, 6, 7, 8, 9, 10.

3, 4, 5, 6, 7, 8, 9, 10.

Não são consideradas sequências válidas:

1, 79, 34, 54, 65, 88, -50.

45, -35, -68, 6, 25.

Ao final, seu programa irá escrever o primeiro termo da maior sequência encontrada e a quantidade de termos da sequência. Para os exemplos de sequência acima, a saída do seu programa será no seguinte formato:

```
Termo inicial: 3  
Maior sequência: 8
```



Atividade 08

Caçando todos os primos

Construa um programa que receba um número qualquer e que apresente todos os números primos de 1 até o número informado, um abaixo do outro. Números primos são números divisíveis apenas por 1 e por ele mesmo.



Atividade 09

Todos os Números Perfeitos

Construa um programa que receba um número inteiro e que verifica e apresenta todos os números perfeitos de 1 até o número informado. Caso encontre um número que seja perfeito, o programa deve imprimir a mensagem "**X é um número perfeito**", na qual **X** é o número encontrado. Cada mensagem deve ser impressa uma abaixo da outra.

Na matemática, um número perfeito é um número inteiro cuja soma de todos os seus divisores positivos próprios (excluindo ele mesmo) é igual ao próprio número. Por exemplo: o número 6 é perfeito, pois $1+2+3$ é igual a 6.

Ufa! Acabamos mais uma aula de exercícios! Nesta, você necessitou utilizar as estruturas de repetição com aninhamento. Além disso, também utilizou, em algumas delas, a estrutura de controle de fluxo. Essa habilidade será extremamente

importante quando for construir algoritmos para as estruturas de dados homogêneas que você conhecerá nas próximas aulas. Portanto, caso você tenha encontrado alguma dificuldade, converse com seu mediador para poder sanar o quanto antes as suas dúvidas.

Continue realizando seus estudos com dedicação! Sucesso! Até a próxima aula!





Resumo

Nesta aula, você exercitou a construção de algoritmos, utilizando o **aninhamento** das estruturas de repetição.



Referências

Linguagem Potigol: Programação para todos. Disponível em:
<<http://potigol.github.io/>>. Acesso em: 30 ago. 2018.