



Lista de exercícios 3

- 1) Escreva um programa que imprima todos os números inteiros de 1 a N, com N sendo fornecido pelo usuário.
- 2) Escreva um programa que imprima a tabuada de um número N entre 1 e 9. Obs: Verifique se o valor N entrado pelo usuário está entre os valores permitidos.
- 3) Escreva um programa que leia N números inteiros e retorne o valor máximo inserido.
- 4) Escreva um programa que leia N números inteiros e retorne o somatório de todos os valores inseridos.
- 5) Escreva um programa que leia um valor N positivo e retorne o fatorial de N.
- 6) Escreva um programa que retorne a soma de todos os números ímpares positivos menores do um valor N entrado pelo usuário.
- 7) Escreva um programa que receba um número N e mostre todos os divisores de N.
- 8) Escreva um programa que receba 2 números inteiros positivos e calcule o mínimo múltiplo comum.
- 9) Escreva um programa que receba 2 números inteiros positivos e calcule o máximo divisor comum.
- 10) Escreva um programa que receba um número N e informe se N é um número primo ou não.
- 11) Escreva um programa que construa uma progressão aritmética a partir do **valor inicial**, da **razão** e da **quantidade de elementos**. Dica: em uma progressão aritmética, cada elementos é igual ao anterior somado com a razão.

Ex:

Valor Inicial = 2

Razão = 3

Quantidade = 6

Progressão Aritmética = 2,5,8,11,14,17

- 12) Escreva um programa que apresente um menu de opções para cálculo das seguintes operações entre dois números:
 1. adição
 2. subtração
 3. multiplicação
 4. divisão
 5. saída

O programa deve possibilitar ao usuário a escolha da operação desejada, a exibição do resultado e a volta do menu de opções. O programa só termina quando for escolhido a opção de saída (opção 5).

13) Escreva um programa que leia um número inteiro N positivo e desenhe formas geométricas usando asteriscos, como mostrados abaixo para o caso de N = 6.

* ** *** **** ***** *****	***** ***** **** *** ** *	* * * * *
* * * * * *	***** ***** **** *** ** *	* ** *** **** ***** *****
* *	* *	* * * * * * * * * * * *

14) Escreva um programa que leia um número inteiro positivo N e em seguida imprima N linhas do chamado triângulo de Floyd.

Ex:

N = 6

```

1
2      3
4      5      6
7      8      9      10
11     12     13     14     15
16     17     18     19     20     21

```

15) Na matemática, o número harmônico designado por $H(n)$ define-se como sendo a soma da série harmônica:

$$H(n) = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

Faça um programa que leia um valor n inteiro positivo e calcule o valor de $H(n)$.

16) Estudiosos já formularam diversas séries matemáticas que, se levadas ao infinito, podem calcular π de forma precisa em inúmeras casas decimais. Algumas delas são tão complexas que só conseguem ser analisadas por supercomputadores. A de Gregory-Leibniz, por sua vez, é uma das mais simples. Para isso, usa-se a seguinte fórmula:

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \frac{4}{11} + \frac{4}{13} - \frac{4}{15} \dots$$

Somas e subtrações são alternadas com frações de numerador 4 e denominador com números ímpares em sequência. Quanto mais longe for, mais perto este resultado será de π . Escreva um programa que leia do usuário uma quantidade positiva N de parcelas da sequência de Gregory-Leibniz e retorne a aproximação do valor de π .

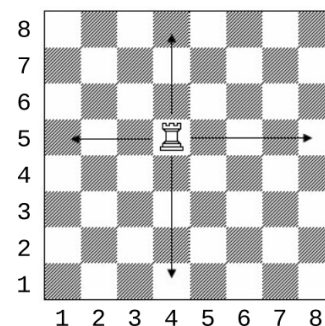
17) A sequência de Fibonacci é uma das sequências matemáticas mais famosas do mundo. A regra de formação dessa sequência é a seguinte: os dois primeiros valores são iguais a 1; e todos os demais elementos são iguais a soma dos dois elementos anteriores. Ex: 1, 1, 2, 3, 5, 8, 13, 21, 34, ... Escreva um programa que leia um número inteiro positivo N e mostre os N primeiros números de Fibonacci.

18) Escreva um programa que leia do usuário um número binário e o converta em um número decimal. (**Obs:** Sem usar função de qualquer biblioteca.) Ex:

Número binário: 1010100

Número decimal: 84

19) No xadrez, a torre se move em linha reta horizontalmente e verticalmente pelo número de casas não ocupadas, até atingir o final do tabuleiro ou ser bloqueado por outra peça, como mostrado na figura ao lado. Construa um programa que receba as coordenadas de uma torre em um tabuleiro vazio e retorne a lista das posições onde a torre pode se mover. Por exemplo, no caso da figura:



Coordenada inicial: (4,5)

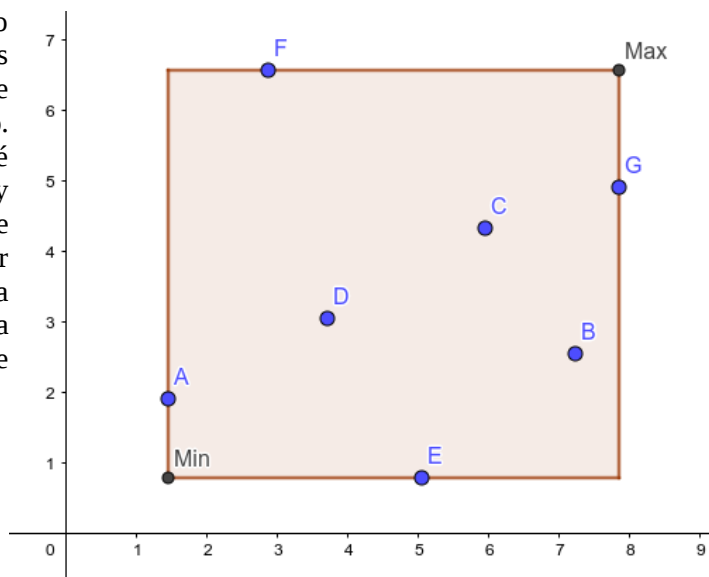
Posições possíveis:

(4,1),(4,2),(4,3),(4,4),(4,6),(4,7),(4,8),(1,5),(2,5),(3,5),(5,5),(6,5),(7,5),(8,5)

20) Escreva um programa que leia como entrada a altura de uma pirâmide de base quadrada como mostrada na figura ao lado e retorne a quantidade de blocos usados na construção da pirâmide.



21) Em muitos jogos é necessário realizar o cálculo de detecção de colisão para saber se dois objetos estão se tocando. Uma etapa deste cálculo é calcular a *bounding box* de um objeto. No caso de duas dimensões, a bounding box é um retângulo alinhado com os eixos x e y definidos pelos pontos Min (extremidade inferior esquerda) e Max (extremidade superior direita) de menor tamanho possível que cubra todos os pontos do objeto. Escreva um programa que leia uma quantidade N de pontos e retorne as coordenadas dos pontos Min e Max.



22) Escreva um programa que leia do usuário as coordenadas de um ponto inicial A, as coordenadas de um vetor de deslocamento b, uma quantidade N e construa a lista de pontos da semirreta partindo de A na direção de b em distâncias regulares do tamanho de b (como indicado na figura abaixo para N = 6).

