1) Desenvolva opção no menu e uma função recursiva para calcular o máximo divisor comum (mdc):
   - mdc(x,y) = y , se x maior e igual y e x módulo y = 0
   - mdc(x,y) = mdc(y,x), se x < y
   - mdc(x,y) = mdc(y, x módulo y), caso contrário.
2) Desenvolva uma opção no meu, para manipular vetores de forma recursiva:
   a) a primeira função deve localizar o menor elemento presente em um vetor de inteiros informado pelo usuário.
   b) a segunda função deve localizar o maior elemento presente em um vetor de inteiros informado pelo usuário.
   c) a terceira função deve calcular a soma de todos os elementos do vetor de inteiro informado pelo usuário.

3) A função de Ackermann nomeada por Wilhelm Ackermann, é um dos mais simples e recém-descobertos exemplos de uma função computável que não são funções recursivas primitivas. Todas as funções recursivas primitivas são totais e computáveis, mas a Função de Ackermann mostra que nem toda função total-computável é recursiva primitiva.
Depois que Ackermann publicou sua função (que continha três inteiros positivos como argumentos), vários autores a modificaram para atender a várias finalidades. Então, a **função de Ackermann** atual pode ser referenciada a uma de suas várias formas da função original. Uma das versões mais comuns, a função de Ackermann-Péter (com dois argumentos), é definida a seguir para os inteiros não negativos *m* e *n*:

$$A(m,n) = \begin{cases} n+1 & se\ m = 0 \\ A(m-1,1) & se\ m > 0\ e\ n = 0 \\ A(m-1, A(m, n-1)) & se\ m > 0\ e\ n > 0 \end{cases}$$

Para ver como a função de Ackermann cresce tão rapidamente, é de grande ajuda expandir algumas expressões simples usando as regras da definição original. Por exemplo, podemos avaliar completamente A (1 , 2) desse jeito:

$$\begin{aligned} A(1,2) &= A(0, A(1,1)) \\ &= A(0, A(0, A(1,0))) \\ &= A(0, A(0, A(0,1))) \\ &= A(0, A(0,2)) \\ &= A(0,3) \\ &= 4. \end{aligned}$$

Para demonstrar como a avaliação de A (4 , 3) resulta em inúmeros passos e em um número gigantesco:

$$
\begin{aligned}
A(4,3) &= A(3, A(4,2)) \\
&= A(3, A(3, A(4,1))) \\
&= A(3, A(3, A(3, A(4,0)))) \\
&= A(3, A(3, A(3, A(3,1)))) \\
&= A(3, A(3, A(3, A(2, A(3,0))))) \\
&= A(3, A(3, A(3, A(2, A(2,1))))) \\
&= A(3, A(3, A(3, A(2, A(1, A(2,0)))))) \\
&= A(3, A(3, A(3, A(2, A(1, A(1,1)))))) \\
&= A(3, A(3, A(3, A(2, A(1, A(0, A(1,0))))))) \\
&= A(3, A(3, A(3, A(2, A(1, A(0, A(0,1))))))) \\
&= A(3, A(3, A(3, A(2, A(1, A(0,2)))))) \\
&= A(3, A(3, A(3, A(2, A(1,3))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(1,2)))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(0, A(1,1))))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(1,0)))))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(0, A(0, A(0,1)))))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(0, A(0,2))))))) \\
&= A(3, A(3, A(3, A(2, A(0, A(0,3)))))) \\
&= A(3, A(3, A(3, A(2, A(0,4))))) \\
&= A(3, A(3, A(3, A(2,5)))) \\
&= \ldots \\
&= A(3, A(3, A(3,13))) \\
&= \ldots \\
&= A(3, A(3,65533)) \\
&= \ldots \\
&= A(3, 2^{65536} - 3) \\
&= \ldots \\
&= 2^{2^{65536}} - 3.
\end{aligned}
$$

Escrito em base 10, esse número seria equivalente à $10^{6.031 \times 10^{19727}}$.

Implemente uma opção que solicite do usuário valores para m e n, e retorne o valor da série de Ackermann.

4) Desenvolva uma opção no meu e uma função recursiva para imprimir os elementos de um vetor de valores inteiros na ordem inversa.

5) Considere a função recursiva 'func' definida por:
func(1) = 1
func(n) = (n - 1) * func(n - 1)

Quais são os valores de func(4) e func(5), respectivamente?

____
____