

Lógica e Algoritmo - T2

Nome: Guilherme Brizzi

1. Escreva um programa que calcule a raiz quadrada aproximada usando o método babilônico, utilize como precisão $\epsilon = 10^{-6}$

início

var: a, b, numero, erroAprox

ler(numero)

a = numero

repita

 b = numero/a

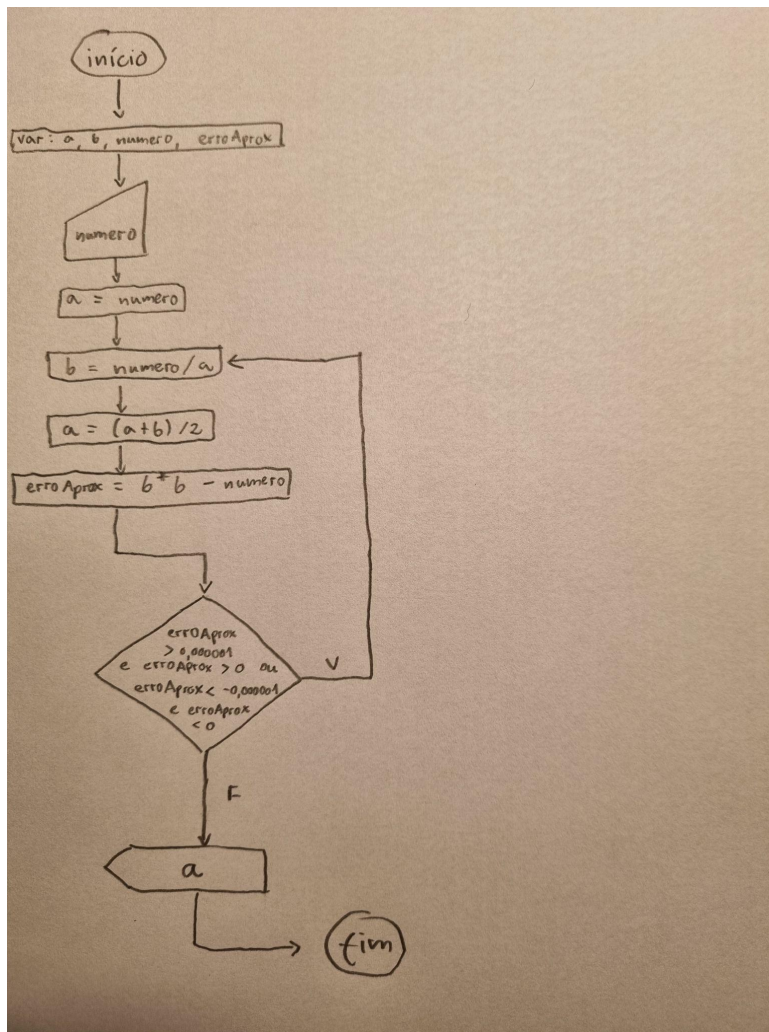
 a = (a + b)/2

 erroAprox = b*b - numero

enquanto erroAprox > 0,000001 e erroAprox > 0 ou erroAprox < -0,000001 e erroAprox < 0

 escreva(a)

fim



2. Escreva um algoritmo que pegue dois números e checa se eles são números amigos ou não.

função divisores(numero)

var: i, somatorio = 0

para i = 1 até numero-1 repita

se numero%i == 0 então

somatorio = somatorio + i

i = i + 1

fim se

fim para

retorne somatório

fim função

início

var: numA, numB

ler(numA, numB)

se divisores(numA) == numB e numA == divisores(numB) então

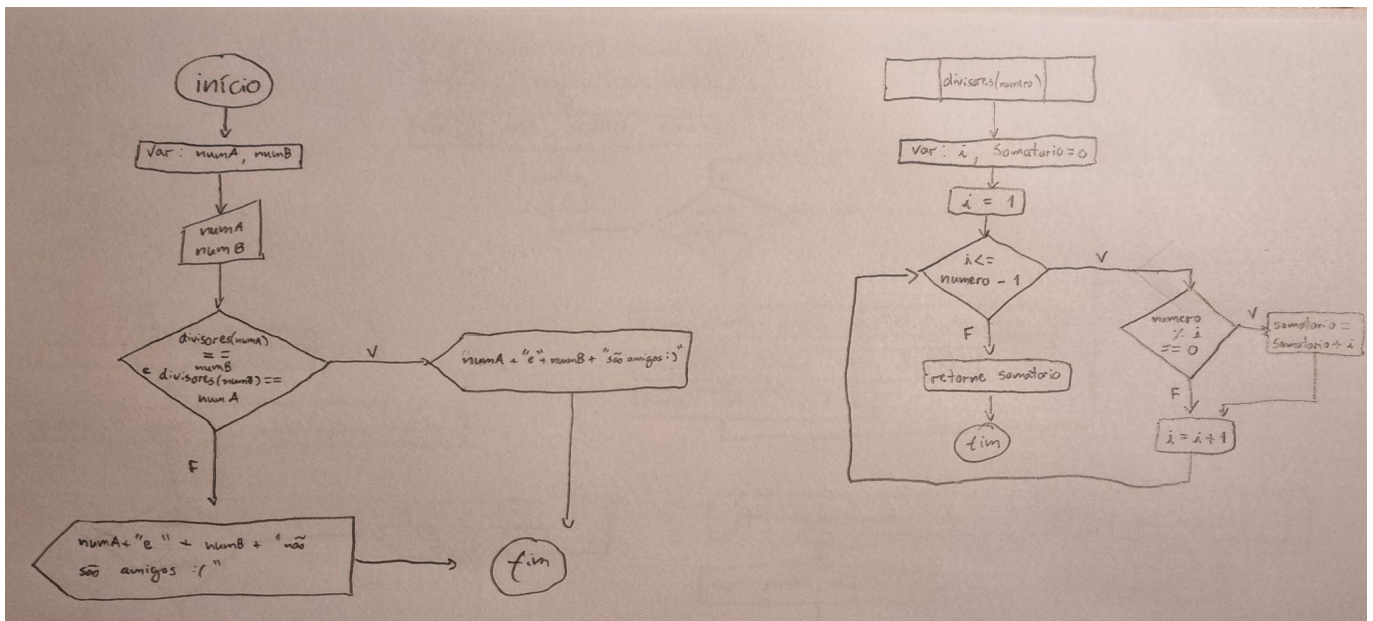
escreva(numA + " e " + numB + " são amigos! :) ")

senão

escreva(numA + " e " + numB + " não são amigos :(")

fim se

fim



3. Escreva um algoritmo que checa se o número dado é um número de Armstrong ou não.

```
função potência(número, expoente)
    var: resultado = 1
    para i = 1 até expoente repita
        i = i + 1
        resultado = resultado * número
    fim para
    retorne resultado
fim função

função qnt_dígitos(número)
    var: divisor, dígitos = 0
    enquanto número >= 1 repita
        número = número / 10
        dígitos = dígitos + 1
    fim enquanto
    retorne dígitos
fim função

função elevar_e_somar(número, dígitos)
    var: i, casa, resultado, soma = 0
    para i = 1 até dígitos repita
        casa = (número / potência(10, i)) % potência(10, i)
        resultado = potência(casa, dígitos)
        soma = soma + resultado
    fim para
    retorne soma
fim função

início
    var: número, dígitos
    ler(número)
    dígitos = qnt_dígitos(número)
    soma = elevar_e_somar(número, dígitos)
    se soma == número então
        escreva("Esse é um número de Armstrong")
    senão
        escreva("Esse não é um número de Armstrong")
    fim se
fim
```

