

# Prova-03: Estrutura de repetição

Prof. Dr. Gustavo Teodoro Laureano

Profa. Dra. Luciana Berretta

Prof. Dr. Thierson Rosa Couto

## Sumário

<b>1</b>	<b>Gerador de tabuada</b>	<b>2</b>
<b>2</b>	<b>Número perfeito</b>	<b>3</b>
<b>3</b>	<b>Cálculo da raiz quadrada</b>	<b>4</b>

# 1 Gerador de tabuada



(++)

Escreva um programa em linguagem C que leia um número qualquer  $n$  de 0 a 9 e imprima na tela a tabuada de soma, subtração, multiplicação e divisão desse número para o  $K$  valores, iniciando em  $i$  em incrementos de  $s$ .

## Entrada

O programa deve ler quatro números quaisquer  $n$ ,  $i$ ,  $K$  e  $s$ .

## Saída

O programa deve apresentar, em sequência, a tabuada de soma, subtração, multiplicação e divisão, com o texto: "Tabuada de soma:", "Tabuada de subtracao:", "Tabuada de multiplicacao:" e "Tabuada de divisao:" antes de cada tabuada. Cada linha da tabuada segue o formato: " $n$  op  $B = R$ ", onde  $n$  é o número lido,  $B$  é o segundo termo da tabuada, op é o operador da tabuada e  $R$  o resultado da operação. Os números devem ser apresentados com 2 casas decimais.

## Exemplo

Entrada
3 1 2 0.1
Saída
Tabuada de soma: 3.00 + 1.00 = 4.00 3.00 + 1.10 = 4.10 Tabuada de subtracao: 3.00 - 1.00 = 2.00 3.00 - 1.10 = 1.90 Tabuada de multiplicacao: 3.00 x 1.00 = 3.00 3.00 x 1.10 = 3.30 Tabuada de divisao: 3.00 / 1.00 = 3.00 3.00 / 1.10 = 2.73

## 2 Número perfeito



(+++)

Dado um número  $n$  inteiro e positivo, dizemos que  $n$  é perfeito se  $n$  for igual à soma de seus divisores positivos diferentes de  $n$ . Construa um programa que leia um número inteiro  $n$ , apresente a soma dos divisores de  $n$  e verifique se o número informado é perfeito ou não.

### Entrada

O programa deve ler um número inteiro  $n$ .

### Saída

O programa deve apresentar uma linha contendo o texto: " $n = d_1 + d_2 + d_3 + \dots + d_k = x$  (MENSAGEM)", onde  $n$  é o número lido,  $d_i$  são os divisores de  $n$  em ordem crescente,  $x$  é a soma dos divisores e MENSAGEM é a mensagem "NUMERO PERFEITO" ou "NUMERO NAO E PERFEITO".

### Observações

Suponha que o usuário sempre fornecerá um número maior que 1.

### Exemplo

Entrada
6
Saída
6 = 1 + 2 + 3 = 6 (NUMERO PERFEITO)

Entrada
12
Saída
12 = 1 + 2 + 3 + 4 + 6 = 16 (NUMERO NAO E PERFEITO)

### 3 Cálculo da raiz quadrada



(+++)

Os Babilônios utilizavam um algoritmo para aproximar uma raiz quadrada de um número qualquer, da seguinte maneira:

Dado um número  $n$ , para calcular  $r = \sqrt{n}$  assume-se uma aproximação inicial  $r_0 = 1$  e calcula-se  $r_i$  para  $i = 1, \dots, \infty$  até que  $r^2 \approx n$ . O algoritmo deve realizar a aproximação enquanto  $|n - r^2| > e$ . O método babilônico é dado pela seguinte equação:

$$r_k = \frac{r_{k-1} + \frac{n}{r_{k-1}}}{2} \quad (1)$$

#### Entrada

O programa deve ler um número **double**  $n$ , cuja raiz quadrada deseja-se obter, e o erro  $e$  que deverá ser considerado pelo algoritmo.

#### Saída

A saída deve apresentar cada iteração do algoritmo, sendo cada linha composta pelo valor aproximado da raiz quadrada de  $n$  com 9 casas decimais, seguido do erro, também com 9 casas decimais.

#### Exemplo

Entrada	
2	
0.00001	
Saída	
r:	1.500000000, err: 0.250000000
r:	1.416666667, err: 0.006944444
r:	1.414215686, err: 0.000006007