

# Prova-01

Prof. Dr. Gustavo Teodoro Laureano  
Prof. Dr. Thierson Rosa Couto

## Sumário

<b>1</b>	<b>Número de finais (++)</b>	<b>2</b>
<b>2</b>	<b>Tempo de jogo em minutos (++)</b>	<b>3</b>
<b>3</b>	<b>Cálculo da raiz quadrada (+++)</b>	<b>4</b>
<b>4</b>	<b>Índices da matriz inferior (+++)</b>	<b>5</b>

# 1 Número de finais (++)



(++)

Em um campeonato de futebol os times são nomeados como Time1, Time2, ..., TimeN. A organização do campeonato deseja saber quais são as finais possíveis dado a quantidade  $N$  de times. Para resolver esse problema, você foi contratado para fazer um programa de computador que, dada a quantidade  $N$  de times, imprima todas as configurações possíveis de finais.

## Entrada

O programa deve ler um número  $N$ , inteiro e positivo, referente à quantidade de times do campeonato.

## Saída

O programa deve apresentar na tela a sequência de finais com cada linha no formato: Final  $k$ : Time $i$  X Time $j$ , onde  $k$  é um contador de finais,  $i$  e  $j$  são as denominações de cada time. Caso o número de times informado for menor que 2, então o programa deve imprimir a mensagem: "Campeonato invalido!".

## Exemplo

Entrada
3
Saída
Final 1: Time1 X Time2
Final 2: Time1 X Time3
Final 3: Time2 X Time3

Entrada
1
Saída
Campeonato invalido!

## 2 Tempo de jogo em minutos (++)



(++)

Faça um programa que leia o horário de início e término de um jogo, em horas e minutos (hora inicial, minuto inicial, hora final, minuto final). Em seguida, imprima a duração do jogo, sabendo que o jogo pode começar em um dia e terminar em outro dia.

### Saída

Quatro números inteiros representando o horário de início e término do jogo.

### Observações

Imprima a duração do jogo em horas e minutos, neste formato: “O JOGO DUROU XXX HORA(S) E YYY MINUTO(S)”. O que significa: o jogo durou XXX hora(s) e YYY minutos. Leia o horário de início e término de um jogo, em horas e minutos (hora inicial, minuto inicial, hora final, minuto final). Em seguida, imprima a duração do jogo, sabendo que o jogo pode começar em um dia e terminar no dia seguinte.

### Exemplo

Entrada	Saída
7 10 8 9	O JOGO DUROU 0 HORA(S) E 59 MINUTO(S)
7 8 9 10	O JOGO DUROU 2 HORA(S) E 2 MINUTO(S)
7 7 7 7	O JOGO DUROU 24 HORA(S) E 0 MINUTO(S)

### 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  $r_k$  para  $k = 1, \dots, \infty$  até que  $r_k^2 \approx n$ . O algoritmo deve realizar a aproximação enquanto  $|n - r_k^2| > e$ . 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

## 4 Índices da matriz inferior (+++)



(+++)

Faça um algoritmo em linguagem C que apresente os pares de índices inferiores à diagonal principal de uma matriz  $m \times n$ . A diagonal principal corresponde aos elementos  $a_{i,i}$ .

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \quad (2)$$

### Entrada

O programa deve ler as dimensões  $m$  e  $n$  da matriz, onde  $m$  é o número de linhas e  $n$  o número de colunas.

### Saída

O programa deve apresentar em cada linha os pares de índices de uma mesma linha. Os pares devem ser apresentados entre parênteses e separados por um ífen.

### Exemplo

Entrada
3
3
Saída
(2, 1)
(3, 1) – (3, 2)

Entrada
6
3
Saída
(2, 1)
(3, 1) – (3, 2)
(4, 1) – (4, 2) – (4, 3)
(5, 1) – (5, 2) – (5, 3)
(6, 1) – (6, 2) – (6, 3)

Entrada
5
2
Saída
(2, 1)
(3, 1) – (3, 2)
(4, 1) – (4, 2)
(5, 1) – (5, 2)