

# Lista-02 - Estruturas de Retetição

Prof. Dr. Gustavo Teodoro Laureano

Profa. Dra. Luciana Berretta

Prof. Dr. Thierson Rosa Couto

## Sumário

<b>1</b>	<b>Fatorial</b>	<b>2</b>
<b>2</b>	<b>Gerador de tabuada</b>	<b>3</b>
<b>3</b>	<b>Hipotenusas inteiras</b>	<b>4</b>
<b>4</b>	<b>Maior segmento crescente de uma sequência</b>	<b>5</b>
<b>5</b>	<b>Maior segmento igual de uma sequência</b>	<b>6</b>
<b>6</b>	<b>N ao cubo</b>	<b>7</b>
<b>7</b>	<b>Número perfeito</b>	<b>8</b>
<b>8</b>	<b>Número primo</b>	<b>9</b>
<b>9</b>	<b>Procura por número amigo</b>	<b>10</b>
<b>10</b>	<b>Série de Taylor para a função cosseno</b>	<b>11</b>
<b>11</b>	<b>Série de Taylor para a função <math>e^x</math></b>	<b>12</b>
<b>12</b>	<b>Série de Taylor para a função seno</b>	<b>13</b>
<b>13</b>	<b>Procura por número amigo</b>	<b>14</b>

# 1 Fatorial



(++)

Dado um número inteiro  $n$ , calcule seu fatorial  $n!$ . O fatorial de um número é dado pela equação:  $n! = n(n-1)(n-2) \dots 1$ . Por definição,  $0! = 1$ .

## Entrada

O programa deve ler um número inteiro  $n$ .

## Saída

O programa deve apresentar uma linha com a mensagem: " $n! = f$ ", onde  $n$  é o número lido e  $f$  o seu fatorial.

## Observações

O fatorial de um número é resultado de uma operação de produtório que pode levar a valores incrivelmente grandes. Lembre-se de usar tipos de dados apropriados ao problema proposto.

## Exemplo

Entrada
2
Saída
2! = 2

Entrada
4
Saída
4! = 24

## 2 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

### 3 Hipotenusas inteiras



(+++)

(IME-USP) Dado um número inteiro positivo  $n$ , determinar todos os inteiros entre 1 e  $n$  que são comprimento da hipotenusa de um triângulo retângulo com catetos inteiros.

#### Entrada

O programa deve ler um valor inteiro  $n$  maior que zero.

#### Saída

O programa deve apresentar uma linha com o texto: "hipotenusa =  $h$ , catetos  $c_1$  e  $c_2$  n", onde  $h$  é uma hipotenusa inteira,  $c_1$  e  $c_2$  são seus catetos inteiros.

#### Exemplo

Entrada
5
Saída
hipotenusa = 5, catetos 3 e 4

Entrada
15
Saída
hipotenusa = 5, catetos 3 e 4
hipotenusa = 10, catetos 6 e 8
hipotenusa = 13, catetos 5 e 12
hipotenusa = 15, catetos 9 e 12

## 4 Maior segmento crescente de uma sequência



(++) (POLI 89) Dados  $n$  e uma sequência de  $n$  números inteiros, determinar o comprimento de um segmento crescente de comprimento máximo.

### Entrada

O programa deve ler um número inteiro maior que zero  $n$  e uma sequência de  $n$  números inteiros em qualquer ordem.

### Saída

O programa deve apresentar a mensagem "O comprimento do segmento crescente maximo e:  $k$ ", onde  $k$  é o tamanho do maior segmento crescente encontrado.

### Exemplo

<b>Entrada</b>
9 5 10 3 2 4 7 9 8 5
<b>Saída</b>
O comprimento do segmento crescente maximo e: 4
<b>Entrada</b>
5 10 8 7 5 2
<b>Saída</b>
O comprimento do segmento crescente maximo e: 1

## 5 Maior segmento igual de uma sequência



(++) (POLI 87) Dados  $n$  e uma sequência de  $n$  números inteiros, determinar quantos segmentos de números iguais consecutivos compõem essa sequência.

### Entrada

O programa deve ler um número inteiro maior que zero  $n$  e uma sequência de  $n$  números inteiros em qualquer ordem.

### Saída

O programa deve apresentar a mensagem "O comprimento do segmento de numeros iguais e:  $k$ ", onde  $k$  é o tamanho do maior segmento crescente encontrado.

### Exemplo

<b>Entrada</b>
2
1 1
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 2
<b>Entrada</b>
7
0 0 2 5 5 5 6
<b>Saída</b>
O comprimento do segmento de numeros iguais e: 3

## 6 N ao cubo



(+++)

(IME-USP) Sabe-se que um número da forma  $n^3$  é igual a soma de  $n$  ímpares consecutivos.

Exemplo:  $1^3 = 1$ ,  $2^3 = 3 + 5$ ,  $3^3 = 7 + 9 + 11$  e  $4^3 = 13 + 15 + 17 + 19$ . Dado  $m$ , determine os ímpares consecutivos cuja soma é igual a  $n^3$  para  $n$  assumindo valores de 1 a  $m$ .

### Entrada

O programa deve ler um número inteiro maior que zero.

### Saída

O programa deve apresentar  $m$  linhas com a seguinte mensagem: " $k * k * k = x_1 + x_2 + \dots + x_k$ ", onde  $k = 1, 2, \dots, m$  e  $x_i$  é a sequência de números ímpares consecutivos.

### Exemplo

Entrada
4
Saída
$1 * 1 * 1 = 1$
$2 * 2 * 2 = 3 + 5$
$3 * 3 * 3 = 7 + 9 + 11$
$4 * 4 * 4 = 13 + 15 + 17 + 19$

## 7 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

<b>Entrada</b>
6
<b>Saída</b>
6 = 1 + 2 + 3 = 6 (NUMERO PERFEITO)

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



## 8 Número primo



(+)

Faça um programa que leia um número  $N$  e informa se o número é primo ou não.

### Entrada

O programa deverá ler um número inteiro  $N$  positivo.

### Saída

O programa deverá apresentar a mensagem "PRIMO" caso  $N$  seja primo e "NAO PRIMO" caso contrário. Caso o valor de  $N$  não seja um número inteiro positivo, o programa deve apresentar a mensagem "Numero invalido.".

### Exemplo

Entrada
7
Saída
PRIMO

Entrada
9
Saída
NAO PRIMO

## 9 Procura por número amigo



(++++)

Números amigos são números onde cada um deles é a soma dos divisores do outro. Por exemplo, o par (220,284) são números amigos porque a soma dos divisores de 220 ( 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110) é igual a 284 e a soma dos divisores de 284 (1, 2, 4, 71 e 142) é igual a 220. Faça um programa que encontre os  $n$  primeiros números amigos do conjunto dos números naturais. O programa deve encontrar somente números amigos diferentes. Por exemplo, o par (220,284) tem o par de números amigos correspondente (284,220), no entanto, o par é formado pelos mesmos números. O programa deve apresentar somente o primeiro par (220,284), de modo que o primeiro número amigo sempre é menor que o segundo.

### Entrada

O programa deve ser um número inteiro positivo  $n$ .

### Saída

Os pares de números devem ser apresentados em linhas separadas, entre parênteses, separados por vírgula e sem espaços entre si. Ex: "(x,y)".

### Observações

A procura por números amigos pode demorar muito tempo. Limite seus testes para  $n < 9$ .

### Exemplo

Entrada
2
Saída
(220,284)
(1184,1210)

## 10 Série de Taylor para a função cosseno



(++++)

Escreva um programa que dado um número real  $x$  e a quantidade de termos  $N$ , calcule o valor da função  $\cos(x)$ , a partir da série:

$$\cos(x) = \sum_{n=0}^N \frac{(-1)^n x^{2n}}{(2n)!} = \frac{x^0}{0!} - \frac{x^2}{2!} + \frac{x^4}{4!} - \dots + \frac{(-1)^N x^{2N}}{(2N)!} \quad (1)$$

, onde  $x$  é o ângulo em radianos e  $N$  a quantidade de termos da série menos 1.

### Entrada

O programa deve ler o valor de  $x$  e  $N$ .

### Saída

O programa deve apresentar uma linha contendo o texto " $\cos(x) = y$ ", onde  $x$  é o ângulo fornecido pelo usuário e  $y$  o cosseno do ângulo.  $x$  deve ser impresso com 2 casas decimais e  $y$  com 6 casas decimais.

### Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de  $x$ . Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de  $N$  maiores que 9. Lembre-se que um ângulo qualquer sempre pode ser representado por um valor entre 0 e  $2\pi$ . Use a constante `M_PI` da biblioteca `<math.h>`. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

### Exemplo

<b>Entrada</b>
2
9
<b>Saída</b>
<code>cos (2.00) = -0.416147</code>
<b>Entrada</b>
3.14
6
<b>Saída</b>
<code>cos (3.14) = -0.999899</code>
<b>Entrada</b>
1
4
<b>Saída</b>
<code>cos (1.00) = 0.540303</code>

## 11 Série de Taylor para a função $e^x$



(++++)

Escreva um programa que dado um número real  $x$  e a quantidade de termos  $N$ , calcule o valor da função  $e^x$ , a partir da série:

$$e^x = \sum_{n=0}^N \frac{x^n}{(n)!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \dots + \frac{x^N}{(N)!} \quad (2)$$

, onde  $x$  é o expoente da função e  $N$  a quantidade de termos da série menos 1.

### Entrada

O programa deve ler o valor de  $x$  e  $N$ .

### Saída

O programa deve apresentar uma linha contendo o texto " $e^x = y$ ", onde  $x$  é o expoente fornecido pelo usuário e  $y$  o valor da função.  $x$  deve ser impresso com 2 casas decimais e  $y$  com 6 casas decimais.

### Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de  $x$ . Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de  $N$  maiores que 9. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

### Exemplo

Entrada
2
9
Saída
$e^{2.00} = 7.388713$

Entrada
3.14
6
Saída
$e^{3.14} = 22.155058$

Entrada
1
9
Saída
$e^{1.00} = 2.718282$

## 12 Série de Taylor para a função seno



(++++)

Escreva um programa que dado um número real  $x$  e a quantidade de termos  $N$ , calcule o valor da função  $\sin(x)$ , a partir da série:

$$\sin(x) = \sum_{n=0}^N \frac{(-1)^n x^{2n+1}}{(2n+1)!} = \frac{x^1}{0!} - \frac{x^3}{3!} + \frac{x^5}{5!} - \dots + \frac{(-1)^N x^{2N+1}}{(2N+1)!} \quad (3)$$

, onde  $x$  é o ângulo em radianos e  $N$  a quantidade de termos da série menos 1.

### Entrada

O programa deve ler o valor de  $x$  e  $N$ .

### Saída

O programa deve apresentar uma linha contendo o texto "seno(x) = y\n", onde  $x$  é o ângulo fornecido pelo usuário e  $y$  o seno do ângulo.  $x$  deve ser impresso com 2 casas decimais e  $y$  com 6 casas decimais.

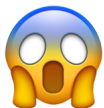
### Observações

Neste tipo de problema, a quantidade de termos pode gerar números muito grandes por conta da operação de fatorial e potenciação de  $x$ . Atente-se aos tipos de dados usados nas declarações das variáveis e não use valores de  $N$  maiores que 9. Lembre-se que um ângulo qualquer sempre pode ser representado por um valor entre 0 e  $2\pi$ . Use a constante `M_PI` da biblioteca `<math.h>`. Como sugestão de desafio à solução do problema, tente escrever um algoritmo que use apenas um laço de repetição.

### Exemplo

<b>Entrada</b>
2
9
<b>Saída</b>
seno(2.00) = 0.909297
<b>Entrada</b>
3.14
6
<b>Saída</b>
seno(3.14) = 0.001614
<b>Entrada</b>
1
4
<b>Saída</b>
seno(1.00) = 0.841471

## 13 Procura por número amigo



(++++)

Números amigos são números onde cada um deles é a soma dos divisores do outro. Por exemplo, o par (220,284) são números amigos porque a soma dos divisores de 220 ( 1, 2, 4, 5, 10, 11, 20, 22, 44, 55 e 110) é igual a 284 e a soma dos divisores de 284 (1, 2, 4, 71 e 142) é igual a 220. Faça um programa que encontre os  $n$  primeiros números amigos do conjunto dos números naturais. O programa deve encontrar somente números amigos diferentes. Por exemplo, o par (220,284) tem o par de números amigos correspondente (284,220), no entanto, o par é formado pelos mesmos números. O programa deve apresentar somente o primeiro par (220,284), de modo que o primeiro número amigo sempre é menor que o segundo.

### Entrada

O programa deve ser um número inteiro positivo  $n$ .

### Saída

Os pares de números devem ser apresentados em linhas separadas, entre parênteses, separados por vírgula e sem espaços entre si. Ex: "(x,y)".

### Observações

A procura por números amigos pode demorar muito tempo. Limite seus testes para  $n < 9$ .

### Exemplo

Entrada
2
Saída
(220,284)
(1184,1210)