# Lista de Repetição

Prof. Msc. Elias Batista Ferreira
Prof. Dr. Gustavo Teodoro Laureano
Profa. Dra. Luciana Berretta
Prof. Dr. Thierson Rosa Couto

## Sumário

# 1 Conversão de temperatura (+)

(+)

Escreva um programa que imprima uma tabela de conversão de graus Fahrenheit para graus Celsius. Dado um valor de temperatura $F$ medida na escala Fahrenheit, seu valor equivalente $C$ na escala Celsius é dado pela seguinte equação:

$$C = \frac{5(F-32)}{9}$$

.

### Entrada

A entrada conterá várias linhas. A primeira delas contém o número $n$ de temperaturas em Fahrenheit a serem convertidas para Celsius. Cada uma das $n$ linhas seguintes contém um valor real (`double`) com a medida de uma temperatura em graus Fahrenheit.

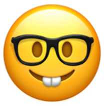
### Saída

O programa deve imprimir $n$ linhas cada uma no seguinte formato $x$ FAHRENHEIT EQUIVALE A $y$ CELSIUS, onde $x$ corresponde a um valor de temperatura em Fahrenheit e $y$ corresponde ao valor equivalente em graus Celsius. Logo após a palavra CELSIUS em cada linha de saída deve ser impresso o caractere de quebra de linha. Os valores de $x$ e $y$ devem ser impressos com duas casas decimais.

### Exemplo

| **Entrada** |
|---|
| `3`<br>`8`<br>`60`<br>`-20` |
| **Saída** |
| `8.00 FAHRENHEIT EQUIVALE A -13.33 CELSIUS`<br>`60.00 FAHRENHEIT EQUIVALE A 15.56 CELSIUS`<br>`-20.00 FAHRENHEIT EQUIVALE A -28.89 CELSIUS` |

# 2 Quadrado de pares (+)

 (+)

[1]

Escreva um programa para ler um valor inteiro $N$ e que gere o quadrado de cada um dos valores pares, de 1 até $N$, inclusive $N$, se for o caso.

### Entrada

A entrada conterá uma linha com um valor inteiro $N$, $5 < N < 2000$.

### Saída

A saída deve conter, uma linha para cada quadrado computado. Em cada linha deve constar uma expressão do tipo $x\hat{}2 = y$, onde $x$ é um número par e $y$ é o seu valor elevado ao quadrado. Imediatamente após o valor de $y$ deve aparecer o caractere de quebra de linha: '\n'.

### Exemplo

| **Entrada** |
|---|
| 6 |
| **Saída** |
| 2^2 = 4<br>4^2 = 16<br>6^2 = 36 |

[1]Fonte: Site do URI - https://www.urionlinejudge.com.br/judge/pt/problems/view/1073.

# 3 Série de pares (+)

(+)

Escreva um programa para ler uma linha com dois números inteiros $x$ e $y$. O programa deve verificar se $x$ é um número par. Se for, o programa deve imprimir uma sequência de $y$ números pares, iniciando com $x$. Se $x$ não for par, o programa deve imprimir uma linha com a mensagem: O PRIMEIRO NUMERO NAO E PAR.

### Entrada

A entrada conterá uma linha com dois números inteiros separados entre si por um caractere de espaço. Após o segundo número na entrada há um caractere de quebra de linha (\n).

### Saída

Se o primeiro número for par, o programa deve imprimir uma linha contendo a sequência de números pares, com um espaço entre cada número par. Após o último número da serie, o programa deve imprimir um espaço seguido de um caractere de quebra de linha ('\n'). Se o primeiro número não for par, o programa deve imprimir a mensagem O PRIMEIRO NUMERO NAO E PAR e logo em seguida, o caractere de quebra de linha.

### Exemplo

| **Entrada** |
|---|
| `20 10` |
| **Saída** |
| `20 22 24 26 28 30 32 34 36 38` |

| **Entrada** |
|---|
| `3 20` |
| **Saída** |
| `O PRIMEIRO NUMERO NAO E PAR` |

# 4 Somatório simples (+)

(+)

Faça um programa que leia um valor $n$, inteiro e positivo, calcule e mostre a seguinte soma:

$$S = \sum_{k=1}^{n} = \frac{1}{k} = 1 + 1/2 + 1/3 + 1/4 + \ldots + 1/n \tag{1}$$

### Entrada

O programa deve ler um número inteiro positivo e maior que 1.

### Saída

O programa deve apresentar uma linha contendo o valor final do somatório com 6 casas decimais. Caso o número lido não atenda as especificações da entrada, o programa deve apresentar a mensagem: "Numero invalido!".

### Observações

Use precisão dupla para o cálculo de $S$.

### Exemplo

| **Entrada** |
|---|
| `10` |
| **Saída** |
| `2.928968` |

# 5 Fatorial (++)

😀 (++)

Dado um número inteiro $n$, calcule seu fatorial $n!$. O fatorial de um número é dado pela equação: $n! = n(n-1)(n-2)\ldots 1$. 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` |

# 6 Maior segmento crescente de uma sequência (++)

(++) (POLI 89) Dados $n$ e uma seqüência 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
n", onde $k$ é o tamanho do maior segmento crescente encontrado.

### Exemplo

| **Entrada** |
|---|
| `9`<br>`5 10 3 2 4 7 9 8 5` |
| **Saída** |
| `O comprimento do segmento crescente maximo e: 4` |

| **Entrada** |
|---|
| `5`<br>`10 8 7 5 2` |
| **Saída** |
| `O comprimento do segmento crescente maximo e: 1` |

# 7 Média de pares e ímpares (++)

😀 (++)

Faça um programa que leia uma sequência de números inteiros diferente de zero e apresente a média nos números pares e a média nos números ímpares.

### Entrada

O programa deve ler uma sequência de números inteiros diferentes de zero.

### Saída

O programa deve apresentar duas linhas, a primeira contendo a mensagem: "MEDIA PAR = *mp*"e a segunda com a mensagem: "MEDIA IMPAR = *mi*", onde *mp* e *mi* são os valores das médias dos números pares e ímpares respectivamente.

### Exemplo

| Entrada | Saída |
|---|---|
| `1 5 8 7 6 3 0` | `MEDIA PAR: 7.000000`<br>`MEDIA IMPAR: 4.000000` |

# 8 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

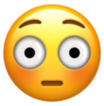
O programa deve apresentar na tela a sequência de finais com cada linha no formato: Final $k$: Time$i$ X Time$j$, onde $k$ é um contador de finais, $i$ e $j$ são as denominações de cada time. Caso o número de times informado for menor que 2, então o programa deve imprimir a mensagem: "Campeonato invalido!".

### Exemplo

| **Entrada** |
|---|
| `3` |
| **Saída** |
| `Final 1:  Time1 X Time2`<br>`Final 2:  Time1 X Time3`<br>`Final 3:  Time2 X Time3` |

| **Entrada** |
|---|
| `1` |
| **Saída** |
| `Campeonato invalido!` |

# 9 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 $k = 1, \ldots, \infty$ 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:

$$r_k = \frac{r_{k-1} + \frac{n}{r_{k-1}}}{2} \tag{2}$$

### 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`<br>`0.00001` |
| **Saída** |
| `r:  1.500000000, err:  0.250000000`<br>`r:  1.416666667, err:  0.006944444`<br>`r:  1.414215686, err:  0.000006007` |

# 10 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 $a_{i,i}$.

$$\begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix} \tag{3}$$

### 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.

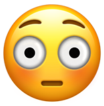
### Exemplo

| **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)` |

# 11 José (+++)

 (+++)

João tem um irmão mais novo, José, que começou a ir à escola e já está tendo problemas com números. Para ajudá-lo a pegar o jeito com a escala numérica, sua professora escreve dois números de três dígitos e pede a José para comparar esses números. Mas em vez de interpretá-los com o dígito mais significativo à esquerda, ele deve interpretá-lo com o dígito mais significativo à direita. Ele tem que dizer à professora qual o maior dos dois números. Escreva um programa que irá verificar as respostas de José.

### Entrada

A entrada conterá um inteiro $T$ , o número de casos de testes, e, para cada caso de teste, uma única linha com dois números de três dígitos, $A$ e $B$, os quais não serão iguais e não conterão zeros.

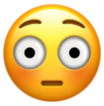
### Saída

A saída deve conter, numa linha para cada caso de teste, com o maior dos números na entrada, comparados como descrito no enunciado da tarefa. O número deve ser escrito invertido, para mostrar a José como ele deve lê-lo.

### Exemplo

| **Entrada** |
|---|
| `3`<br>`734 893`<br>`221 231`<br>`839 237` |
| **Saída** |
| `437`<br>`132`<br>`938` |

# 12 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. Para cada valor de hipotenusa válido no intervalo de 1 a $n$, imprimir todos os pares de catetos que formam um triângulo retângulo distinto com aquele valor de hipotenusa.

### 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 $c_2$", onde $h$ é uma hipotenusa inteira, $c_1$ e $c_2$ são seus catetos inteiros, de modo que $c_1 \leq c_2$. No caso de haver mais de um par de catetos válidos para um mesmo valor de hipotenusa, por exemplo $(c1, c2), (c3, c4), \ldots (c_k, c_{k+1})$, imprima os pares de tal modo que o valor do primeiro cateto seja menor ou igual ao valor do segundo cateto de um mesmo par e que o valor do primeiro cateto de um par seja menor que o valor do primeiro cateto do par de subsequente. Por exemplo, para um valor de hipotenusa igual a 85, existem os seguintes pares de catetos: $(13, 84)$, $(40, 75), (36, 77) e (51, 68)$. Nesse caso a saída deve ser a seguinte:

hipotenusa = 85, catetos 13 e 84
hipotenusa = 85, catetos 36 e 77
hipotenusa = 85, catetos 40 e 75
hipotenusa = 85, catetos 51 e 68

### Exemplo

| **Entrada** |
|---|
| 5 |
| **Saída** |
| `hipotenusa = 5, catetos 3 e 4` |

| **Entrada** |
|---|
| 15 |
| **Saída** |
| `hipotenusa = 5, catetos 3 e 4`<br>`hipotenusa = 10, catetos 6 e 8`<br>`hipotenusa = 13, catetos 5 e 12`<br>`hipotenusa = 15, catetos 9 e 12` |

# 13 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: "$k*k*k = x_1 + x_2 + \ldots + x_k$", onde $k = 1, 2, \ldots, m$ e $x_i$ é a sequência de números ímpares consecutos.

### Exemplo

| **Entrada** |
| --- |
| `4` |
| **Saída** |
| `1*1*1 = 1`<br>`2*2*2 = 3+5`<br>`3*3*3 = 7+9+11`<br>`4*4*4 = 13+15+17+19` |

# 14 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 $n$, apresenta a soma dos divisores de $n$ e verifica 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 = d1 + d2 + d3 + ... + dk = 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

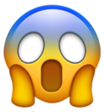
Suponha que o usuário sempre fornecerá um número maior que 1.

### 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)` |

# 15 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 pos vírgula e sem espaços entre si. Ex: "(x,y)".

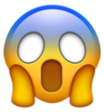
### Observações

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

### Exemplo

| **Entrada** |
|---|
| `2` |
| **Saída** |
| `(220,284)`<br>`(1184,1210)` |

# 16 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:

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

, 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\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.

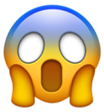
#### Exemplo

| **Entrada** |
|---|
| `2`<br>`9` |
| **Saída** |
| `cos(2.00) = -0.416147` |

| **Entrada** |
|---|
| `3.14`<br>`6` |
| **Saída** |
| `cos(3.14) = -0.999899` |

| **Entrada** |
|---|
| `1`<br>`4` |
| **Saída** |
| `cos(1.00) = 0.540303` |

# 17 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:

$$e^x = \sum_{n=0}^{N} \frac{x^n}{(n)!} = \frac{x^0}{0!} + \frac{x^1}{1!} + \frac{x^2}{2!} + \ldots + \frac{x^N}{(N)!} \tag{5}$$

, 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\n", 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.

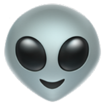
### Exemplo

| **Entrada** |
|---|
| `2`<br>`9` |
| **Saída** |
| `e^2.00 = 7.388713` |

| **Entrada** |
|---|
| `3.14`<br>`6` |
| **Saída** |
| `e^3.14 = 22.155058` |

| **Entrada** |
|---|
| `1`<br>`9` |
| **Saída** |
| `e^1.00 = 2.718282` |

# 18 Decomposição em fatores primos (+++++)

 (+++++)

Todo número natural maior que 1 pode ser escrito na forma de uma multiplicação em que todos os fatores são números primos. Por exemplo, o número 36 pode ser representado pela multiplicação $2 \times 2 \times 3 \times 3$. A essa representação multiplicativa dá-se o nome de Decomposição em Fatores Primos ou Fatoração, que é um produto de fatores primos. O processo de fatoração de $N$ segue um método prático de divisões sucessivas pelo seu menor fator primo. A cada passo, deve-se encontrar o menor divisor primo do quociente da divisão anterior. A Figura 1 mostra dois exemplos d efatoração em números primos.

Faça um programa que leia um número inteiro maior que 1 e apresente sua fatoração em números primos. Uma vez executado, o programa deve sempre apresentar uma fatoração. Caso o número lido seja inválido, o programa deve lê-lo novamente.

| 36 | 2 |
|---|---|
| 18 | 2 |
| 9 | 3 |
| 3 | 3 |
| 1 | |

| 120 | 2 |
|---|---|
| 60 | 2 |
| 30 | 2 |
| 15 | 3 |
| 5 | 5 |
| 1 | |

Figura 1: Exemplo de fatoração dos números 36 e 120.

### Entrada

O programa deve ler um número inteiro $N$.

### Saída

O programa deve apresentar a mensagem "Fatoracao nao e possivel para o numero x!"sempre que o número lido não é válido. Caso o número lido seja válido, então o programa deve apresentar sua fatoração no seguinte formato: `N = f1 x f2 x ... x fk.`

### Exemplo

| **Entrada** |
|---|
| `554` |
| **Saída** |
| `554 = 2 x 277` |

| **Entrada** |
|---|
| `-1`<br>`0`<br>`120` |
| **Saída** |
| `Fatoracao nao e possivel para o numero -1!`<br>`Fatoracao nao e possivel para o numero 0!`<br>`120 = 2 x 2 x 2 x 3 x 5` |

| **Entrada** |
|---|
| `2` |
| **Saída** |
| `2 = 2` |