

# Universidade Federal de Goiás

## Instituto de Informática

### Introdução à Programação - 2017-1

### Lista de Exercícios 1 (L1)

#### Professores:

Gustavo Teodoro Laureano

Luciana de Oliveira Beretta





Thierson Couto Rosa

Tarefa L1-C

## Introdução

A primeira lista de exercícios - Lista de exercícios 1 (L1) contém 30 exercícios. Essa lista está dividida em três tarefas (*assignments*) no sistema Sharif. Essa é a terceira tarefa (L1-C) e contém os oito últimos exercícios de L1. Todos os exercícios de L1-C devem ser resolvidos individualmente. Para cada exercício deve ser escrito um programa, o qual deve ser submetido para avaliação no sistema Sharif<sup>1</sup>. Os exercícios de L1 são classificados em quatro níveis de dificuldade sob o ponto de vista de um aluno iniciante: fácil, intermediário, difícil e muito difícil. Cada grau de dificuldade é indicado por um emoticon distinto, colocado ao lado direito do título do exercício. O valor de cada nível é mostrado na tabela a seguir:

Tabela 1: Valor de cada nível de exercício

Nível	Emoticon	Valor
Fácil		0,3
Intermediário		0,5
Difícil		1,0
Muito Difícil		2,0

A lista L1 vale 10 pontos. O aluno pode conseguir somar 10 pontos de várias maneiras, resolvendo uma combinação de exercícios dos quatro níveis espalhados entre as três tarefas que formam a L1. Embora não seja necessário a resolução de todos os exercícios da lista L1, recomenda-se, principalmente aos iniciantes em programação, que tentem resolver o maior número de exercícios possível. O treinamento é muito importante para desenvolver o raciocínio lógico na solução de problemas.

---

<sup>1</sup><http://sharif.inf.ufg.br/thierson/index.php/login>.

# Conversões de Graus Fahrenheit para Graus Celsius 😊

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 (float) 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.

## Exemplos

Entrada
3
8
60
-20

Saída:
8.00 FAHRENHEIT EQUIVALE A -13.33 CELSIUS
60.00 FAHRENHEIT EQUIVALE A 15.56 CELSIUS
-20.00 FAHRENHEIT EQUIVALE A -28.89 CELSIUS

# Progressão Aritmética Sem Fórmula 😊

Nerdilton adora Matemática, mas ele não tem a memória muito boa para memorizar fórmulas. Geralmente ele entende como a fórmula funciona e deriva o valor de saída da fórmula. Ele precisa responder a um conjunto de exercícios que pedem a soma dos  $n$  primeiros elementos de uma progressão aritmética, dados o primeiro elemento da progressão e a razão. Escreva um programa que seja capaz de encontrar a soma dos  $n$  primeiros termos de uma progressão aritmética sem usar a fórmula fechada da soma dos elementos de uma progressão aritmética. Lembre-se que uma progressão aritmética com valor inicial  $a_1$  e razão  $r$  é a sequência formada por:  $a_1, a_1 + r, a_1 + 2r, \dots, a_1 + (n - 1)r$ . O programa precisa computar a soma dos valores dessa sequência.

## Entrada

A entrada conterá uma linha com três números inteiros separados entre si por um caractere de espaço. O primeiro número corresponde ao valor inicial da progressão aritmética, o segundo, corresponde à razão da mesma e o terceiro corresponde ao número  $n$  de elementos da progressão.

## Saída

O programa deve imprimir o valor da soma dos  $n$  primeiro elementos da progressão, seguido de um caractere de quebra de linha.

## Exemplos

A seguir são mostrados dois casos distintos de entrada, somente para efeito de ilustração, porém, esse problema contém apenas um caso de teste na entrada, formado por apenas uma linha.

Entrada	Saída:
2 4 5	50
7 -2 10	-20

## 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 conterà 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.

### Exemplos

A seguir são mostrados dois casos distintos de entrada, somente para efeito de ilustração, porém, esse problema contém apenas um caso de teste na entrada, formado por apenas uma linha.

Entrada	Saída:
20 10	20 22 24 26 28 30 32 34 36 38
3 20	O PRIMEIRO NUMERO NAO E PAR

# Ultrapassagem entre Populações 😊

Supondo que a população de um país  $A$  seja de  $a$  habitantes com uma taxa anual de crescimento de 3% e que a população de um país  $B$  seja de  $b$  habitantes, com uma taxa anual de crescimento de 1,5%, fazer um algoritmo que calcule e escreva o número de anos necessários para que a população do país  $A$  ultrapasse ou iguale a população do país  $B$ , mantidas essas taxas de crescimento.

## Entrada

O programa deverá ler duas linhas de entrada, cada uma contendo um número inteiro positivo representando a população de um país. O valor na primeira linha corresponde ao número de habitantes do país  $A$  e será sempre menor que o valor na segunda linha, o qual corresponde ao número de habitantes do país  $B$ .

## Saída

A saída deve conter, numa linha com a frase  $\text{ANOS} = x$ , onde  $x$  é um valor em anos e deve ser seguido por um caractere de quebra de linha: ‘\n’.

## Exemplos

A seguir são mostrados dois casos distintos de entrada, somente para efeito de ilustração, porém, esse problema contém apenas um caso de teste na entrada, formado pelas duas linhas de entrada descritas acima.

Entrada
900000000
2000000000

Saída:
ANOS = 55

# Quadrado de Pares<sup>2</sup> 😊

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 conterà 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  $x^2 = y$ , 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’.

## Exemplos

Entrada
6

Saída:
2^2 = 4
4^2 = 16
6^2 = 36

---

<sup>2</sup>Fonte: Site do URI - <https://www.urionlinejudge.com.br/judge/pt/problems/view/1073>.

# Salário 🤖

Escreva um programa que leia várias linhas contendo cada uma a matrícula de um funcionário (um valor inteiro), seu número de horas trabalhadas, o valor que recebe por hora e calcule o salário desse funcionário. A seguir, mostre o número e o salário do funcionário, com duas casas decimais.

## Entrada

A entrada é formada por várias linhas, cada uma contendo três valores decimais separados um do outro por um espaço em branco. O último valor na linha é seguido pelo caractere de quebra de linha (`\n`). A última linha contém uma matrícula igual a zero e não deve ser processada. Ela serve apenas para indicar ao programa o término da entrada de dados.

## Saída

A saída deve conter para cada linha de entrada a matrícula, um espaço em branco e o salário calculado com duas casas decimais

## Exemplos

Entrada:		
2015001	16000	3.23
2015002	16010	3.0
2015003	16009	2.99
2015004	15080	3.13
0	0	0.0

Saída:	
2015001	51680.00
2015002	48030.00
2015003	47866.91
2015004	47200.40

## Observação

Para ler uma linha com os três valores, utilize a função **scanf**: `scanf("%d %f %f", &A, &B, &C);` e, em seguida, a função `getchar()`. A função `getchar()` é usada para consumir o caractere de quebra de linha na entrada.

## 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:	Saída:
3	437
734 893	132
221 231	938
839 237	



# Seqüências Crescentes 🙄

Tia Zélia é uma professora muito dedicada. Ela está explicando as relações de ordem entre números para seus alunos. Na aula de hoje ela explicou a relação “menor que” representada pelo operador  $<$ . Ela explicou também sobre as propriedades dessa relação entre os números, incluindo a propriedade transitiva, isto é, se  $x < y$  e  $y < z$ , então  $x < z$ . Tia Zélia quer fazer um treinamento com seus alunos. Ela quer propor o seguinte exercício aos seus queridos pupilos: apresentar a eles várias seqüências de números e pedir que eles analisem cada seqüência e que indiquem se ela está na ordem crescente. Para isso, tia Zélia quer sua ajuda. Ela vai editar um arquivo com várias seqüências de números; para cada seqüência haverá duas linhas no arquivo: uma com um número inteiro indicando o tamanho da seqüência e a linha imediatamente seguinte contém a seqüência de valores reais propriamente dita, formada por números separados por um espaço, a menos do último valor que vem seguido diretamente por um caractere de quebra de linha. Ela quer gerar seqüências de números sem ter o trabalho de verificar se formam seqüência ou não. Ela quer que um programa de computador faça isso para ela. Tia Zélia ficou sabendo que você é aluno de Introdução à Programação do INF-UFG e que os alunos dessa disciplina são exímios programadores! Portanto ela pede a você que faça um programa que resolva esse problema para ela.

## Entrada

Para cada seqüência numérica há na entrada duas linhas: uma com, apenas um valor inteiro, indica o número de valores reais que deve ocorrer na próxima linha. A linha seguinte contém tantos valores quanto indicado na linha anterior. Entre dois valores há apenas um espaço e após o último valor há um caractere de quebra de linha. A última linha da entrada contém um tamanho de seqüência igual a zero e serve apenas para indicar término do processamento. Não há uma linha com seqüência de valores após a ocorrência de uma linha com valor zero.

## Saída

Para cada seqüência da entrada o seu programa deve emitir uma das seguintes respostas: ORDENADA, se a seqüência estiver em ordem crescente de valores ou DESORDENADA, em caso contrário. Após cada palavra impressa deve haver apenas um caractere de quebra de linha.

## Exemplos

Entrada:
10
2.98 16.42 18.0 23.67 31.99 38.50 42.30 61.78 2000.00 2000.10
5
4.51 4.32 4.90 56.70 150.80
6
0.00 2.56 4.00 80.4 100.98 100.97
0

Saída:
ORDENADA
DESORDENADA
DESORDENADA