



MARATONA DE PROGRAMAÇÃO

# InterFatecs

XI Maratona InterFatecs  
Santana de Parnaíba 2022

Fase 1 - 28/05/22

Caderno de  
Problemas

Organização e Patrocínio

**Fatec**  
Santana de  
Parnaíba

**CPS**  
Centro  
Paula Souza

**SÃO PAULO**  
GOVERNO DO ESTADO  
| Secretaria de Desenvolvimento Econômico

 **Bay  
Metrics**™

[interfatecs.com.br](http://interfatecs.com.br)

# 1 Instruções

Este caderno contém 9 problemas – identificados por letras de A até I, com páginas numeradas de 3 até 24. Verifique se seu caderno está completo.

Informações gerais

## 1. Sobre a competição

- (a) A competição possui duração de 5 horas (início as 13:00h término as 18:00h);
- (b) NÃO é permitido acesso a conteúdo da Internet ou qualquer outro meio eletrônico digital;
- (c) É permitido somente acesso a conteúdo impresso (cadernos, apostilas, livros);
- (d) Não é permitida a comunicação com o técnico ou qualquer outra pessoa que não seja a equipe para tirar dúvidas sobre a maratona
- (e) Cada equipe terá acesso a 1 computador dotado do ambiente de submissão de programas (BOCA), dos compiladores, link-editores e IDEs requeridos pelas linguagens de programação permitidas;
- (f) NÃO é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
- (g) Os problemas têm o mesmo valor na correção.

## 2. Sobre o arquivo de solução e submissão:

- (a) O arquivo de solução (o programa fonte) deve ter o mesmo nome que o especificado no enunciado (logo após o título do problema);
- (b) confirme se você escolheu a linguagem correta e está com o nome de arquivo correto antes de submeter a sua solução;
- (c) NÃO insira acentos ou outros caracteres especiais no arquivo-fonte.

## 3. Sobre a entrada

- (a) A entrada de seu programa deve ser lida da entrada padrão (não use interface gráfica);
- (b) Seu programa será testado em vários casos de teste válidos além daqueles apresentados nos exemplos. Considere que seu programa será executado uma vez para cada caso de teste.

## 4. Sobre a saída

- (a) A saída do seu programa deve ser escrita na saída padrão;
- (b) Não exiba qualquer outra mensagem além do especificado no enunciado.

## 5. Versões das linguagens

- (a) gcc version 5.4.0 (lembre-se que não existem bibliotecas do Windows)
- (b) g++ version 5.4.0 (lembre-se que não existem bibliotecas do Windows)
- (c) javac 1.8.0-272 (lembre-se que as classes devem estar fora de pacotes)
- (d) Python 2.7.12 e 3.5.2 (lembre-se que não é possível usar f-string)

## Problem A

### License Plates

*Source file:* licenseplates.{ c | cpp | java | py }

*Author:* Prof. Antonio Cesar de Barros Munari (Fatec Sorocaba)

XPTO is the software company responsible for a software that processes vehicle license plates captured by traffic surveillance cameras. Its developers are currently working on a software update required due to the adoption of the new Mercosur license plates, which are replacing the traditional ones - composed of 3 letters followed by 4 numerical digits. Figure 1 shows a license plate in the new format, in which we have 3 letters followed by a numeric digit, another letter, and finally two numeric digits. Figure 2 illustrates the license plate model that are being replaced, where the hyphen that separates the letters from the numeric digits is just a visual element of the layout - it is not part of the plate itself.



Figure 1 – The new license plate model for Mercosur countries.



Figure 2 – Old Brazilian license plate model.

When starting the project, XPTO needs something very basic: given a list of license plates provided by the application that decodes the images captured by the cameras, (i) identify which are the new plates, (ii) which are the plates in the old format, and (iii) the invalid readings that may have occurred. Due to the characteristics of the software that decodes the images, all readings bring only the characters ASCII codes. For instance, the plate in Figure 1 would appear as 65 66 67 49 67 51 52 and the plate in Figure 2 would appear as 88 88 88 48 48 48 48 (without the hyphen).

Your task is to determine, for a given license plate, whether it is a new model, an old model, or an invalid read. Assume that the valid letters will be capital letters only (codes 65 to 90 in the ASCII table). Additionally, remember that the numeric digits correspond to ASCII characters with code 48 to 57.

## Input

The input is composed of many lines with 7 integers separated by a white space each, representing each license plate character as captured by the cameras. Each integer is in the interval [1, 200].

## Output

Print “MERCOSUL” for the new Mercosur plates, “ANTIGA” for the old model plates, and “ERRO” for the inputs that do not fit any of these license plate models.

### Example of Input 1

65	66	67	49	67	51	52
88	88	88	48	48	48	48
23	45	72	23	89	42	43

### Example of Output 1

MERCOSUL
ANTIGA
ERRO

## Problema B

# Brasiliávia

*Arquivo fonte:* brasiliavia.{ c | cpp | java | py }

*Autor:* Prof. Dr. Leandro Luque (Fatec Mogi das Cruzes)

O império da Brasiliávia descobriu recentemente que, devido a um esquema generalizado de corrupção, diversas doses de vacina contra a Covid-19 aplicadas em seus cidadãos tinham apenas água em sua composição. A empresa fornecedora, do namorado da irmã da namorada do irmão da esposa de um deputado do país, ganhou um contrato de emergência sem licitação com a ajuda de um lobista conhecido. Isso explica o grande número de casos da doença nos últimos meses.

Logo após a descoberta, a AIB - Agência de Inteligência da Brasiliávia iniciou um programa de mapeamento dos grupos de risco da Covid-19. Para isso, desenvolveu um sistema que, dados os contatos que as pessoas tiveram nos últimos 14 dias - período de maior risco de transmissão do vírus -, indica quantos e quais grupos de pessoas estão seguros, bem como quantos e quais estão inseguros. Um grupo é um conjunto de pessoas tal que todas elas tiveram alguma interação direta ou indireta entre si nos últimos 14 dias e nenhuma delas teve interação com alguém de fora do grupo. O sistema recebe constantemente informações sobre os contatos estabelecidos. Ainda, médicos estão frequentemente consultando o sistema para obter informações sobre os grupos.

### *Exemplo*

Como exemplo, suponha uma população de 20 pessoas numeradas de 1 a 20 e a seguinte sequência de eventos.

Inicialmente, existem 20 diferentes grupos seguros na população, cada um formado por cada pessoa.

- Pessoa 1 teve contato com 2;
- Pessoa 1 teve contato com 3;
- Pessoa 2 teve contato com 4;

Neste momento, existem 17 grupos seguros na população. Um deles formado por 1, 2, 3 e 4; e outros 16 grupos unitários formados por cada uma das 16 pessoas restantes.

- Pessoa 14 teve contato com 7;

Existem 16 grupos seguros na população. Um deles formado por 1, 2, 3 e 4; outro formado por 7 e 14; além de outros 14 grupos unitários formados por cada uma das 14 pessoas restantes.

- Pessoa 14 confirmada com Covid-19;

Agora, existe 1 grupo inseguro, formado por 7 e 14, além de 15 grupos seguros.

- Pessoa 7 teve contato com 2;

Por fim, existem 15 grupos na população. Um deles inseguro, formado por 1, 2, 3, 4, 7 e 14; além de outros 14 grupos unitários formados por cada uma das 14 pessoas restantes.

### Sua tarefa

Você foi o programador responsável pelo desenvolvimento deste sistema e precisa fornecer parte do código para uma auditoria, conforme a especificação de entrada e saída seguintes.

### Entrada

A primeira linha da entrada contém dois inteiros  $P$  ( $1 \leq P \leq 10.000$ ) e  $E$  ( $1 \leq E \leq 10.000$ ), separados por um espaço em branco, representando, respectivamente, o número de pessoas na população e a quantidade de eventos referentes aos últimos 14 dias que serão informados no sistema. As próximas  $E$  linhas contém um comando  $C$  ( $C \in \{c, p, n, ns, ni, ii\}$ ), seguido, em dois casos, por números inteiros, conforme a especificação seguinte:

- $c \ A \ B$ , indica que as pessoas  $A$  e  $B$  ( $1 \leq A, B \leq P$ ) tiveram contato;
- $p \ D$ , indica que a pessoa  $D$  ( $1 \leq D \leq P$ ) foi confirmada com Covid-19;
- $n$ , requisita a impressão do número de grupos atualmente na população;
- $ns$ , requisita a impressão do número de grupos seguros atualmente na população;
- $ni$ , requisita a impressão do número de grupos inseguros atualmente na população;
- $ii$ , requisita a impressão das pessoas que estão em grupos inseguros atualmente na população.

### Saída

A saída contém diversas linhas com um inteiro  $R$  ou o texto "vazio", indicando a resposta para os comandos  $n$ ,  $ns$ ,  $ni$  e  $ii$ , na ordem em que foram informados. Para o comando  $ii$ , os números dos indivíduos devem ser impressos em ordem crescente e separados por um espaço. Caso não haja nenhum indivíduo inseguro, deve-se imprimir a palavra "vazio", sem aspas e em minúsculas.

#### Exemplo de Entrada 1

```
20 14
c 1 2
c 1 3
c 2 4
n
c 14 7
ns
ni
ii
p 14
ni
ii
c 7 2
ii
ni
```

#### Exemplo de Saída 1

```
17
16
0
vazio
1
7 14
7 14 1 2 3 4
1
```

## Problema C

# PoliCripto

*Arquivo fonte:* policripto.{ c | cpp | java | py }

*Autor:* Prof. Dr. Leandro Luque (Fatec Mogi das Cruzes)

Um grupo político extremista tem procurado formas de se comunicar sem que suas conversas sejam interceptadas. Como são pessoas atrapalhadas, em geral, não se acostumam com a ideia de chaves públicas e privadas. Resolveram então criar um código de comunicação que qualquer um que conheça pode usar, sem a necessidade de chaves.

O código é formado por duas sequências. Uma delas composta apenas por caracteres minúsculos, sem caracteres especiais. A outra por dígitos e os caracteres ‘+’ e ‘-’.

Um exemplos destas sequências é:

*dqnptacwulwupaq*

*-122136 + 201 - 31 + 1 - 2*

Na sequência de caracteres, o ‘w’ exerce um papel especial, sempre representando um espaço. Desta forma, a sequência pode ser inicialmente modificada para “dqnptac ul upaq”.

Os números, por sua vez, indicam para quantos caracteres acima ou abaixo cada caractere da sequência deve ser modificado. Um número positivo N indica que o caractere em questão deve ser modificado pelo N-ésimo caractere após ele, quando se considera a ordem alfabética. Um número negativo N indica que o caractere em questão deve ser modificado pelo N-ésimo caractere anterior a ele. O alfabeto deve ser considerado como circular, ou seja, após o ‘z’ vem o ‘a’ novamente - e vice-versa.

A presença de um caractere ‘-’ na sequência numérica indica que todos os dígitos após ele são considerados negativos. De forma similar, a presença de um caractere ‘+’ indica que todos os dígitos após ele são considerados positivos. No início da sequência numérica, caso não haja nenhum caractere ‘-’, os números são considerados positivos por padrão.

Considerando estas regras e as sequências apresentadas, as seguintes transformações devem ser aplicadas a cada um dos caracteres, resultando na mensagem decodificada: “coloque um robo”.

d	q	n	p	t	a	c	u	l	u	p	a	q
-1	-2	-2	-1	-3	-6	2	0	1	-3	-1	1	-2
c	o	l	o	q	u	e	u	m	r	o	b	o

Você, investigador da Polícia Federal, descobridor das regras usadas na codificação, precisa escrever um programa que, dada uma mensagem, obtém a original.

### Entrada

A entrada é composta por duas linhas. A primeira contém a mensagem codificada como uma sequência de caracteres minúsculos com tamanho máximo 200. A segunda linha contém uma sequência de dígitos e caracteres ‘+’ e ‘-’, com tamanho máximo 400.

## Saída

A saída deve imprimir a mensagem decodificada seguida por uma quebra de linha.

### Exemplo de Entrada 1

```
dqnptacwulwupaq  
-122136+201-31+1-2
```

### Exemplo de Saída 1

```
coloque um robo
```

### Exemplo de Entrada 2

```
vdwgzgzwzcahla fwmfuv  
-3+159013-2+1-390+99989
```

### Exemplo de Saída 2

```
se liga cabecao voce
```



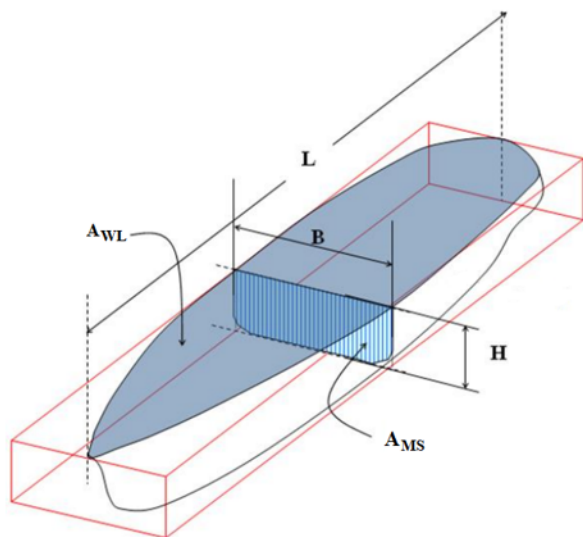
## Problema D

# Seção Mestra

*Arquivo fonte:* secaoomestra.{ c | cpp | java | py }

*Autor:* Prof. Me. Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)

Imagine um navio cortado transversalmente no ponto onde ocorre a sua maior área de seção transversal. Esse ponto ocorre na metade de seu comprimento ou bem perto disso. Chama-se de Seção Mestra justamente essa maior seção transversal, como pode ser visto na Figura 1.



### Glossário

L = Comprimento do Navio

B = Boca – Largura do navio na linha d'água de projeto

H = Calado – Altura da parte submersa do navio

$A_{MS}$  = Área da Seção Mestra (Midship Section Area)

$A_{WL}$  = Área na linha d'água (Waterline Area)

*Figura 1 - Áreas da Seção Mestra e da Linha d'água.*

*fonte:* Pinto, Marcos M. O., *Introdução à Engenharia Naval*

*endereço:* <https://bit.ly/3M8ckKq> acessado em 12/04/2022

Nos estágios iniciais do projeto de um navio existem alguns parâmetros muito significativos que auxiliam e direcionam o desenvolvimento da geometria do casco. Um desses parâmetros é o Coeficiente de Seção Mestra ( $C_M$ ), definido como a razão entre a Área da Seção Mestra ( $A_{MS}$ ) e a área do retângulo calculada pela multiplicação Boca x Calado ( $B \times H$ ), assim:

$$C_M = \frac{A_{MS}}{B \times H}$$

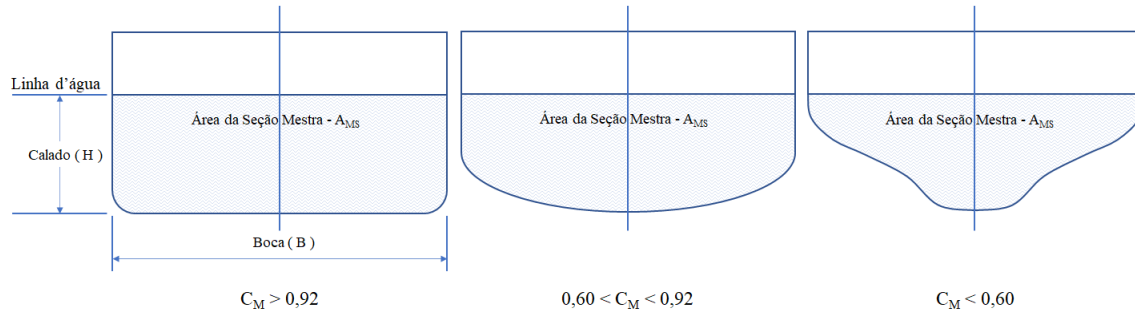


Figura 2 – Casos típicos de Seção Mestra.  
fonte: o autor

A Figura 2 ilustra como o  $C_M$  pode ser útil para dar uma indicação do perfil da embarcação. Em geral ocorre que, quando este coeficiente é alto, o navio tende a ser lento e com grande volume interno (capacidade de carga) e, quando o coeficiente é baixo, o navio é veloz e com menor volume.

Nossos amigos do Curso de Construção Naval da Fatec de Jahu entendem bem desses assuntos e gostariam de poder contar com sua ajuda para fazer um programa capaz de calcular o  $C_M$  a partir de alguns dados básicos aliados a algum conhecimento de cálculo numérico.

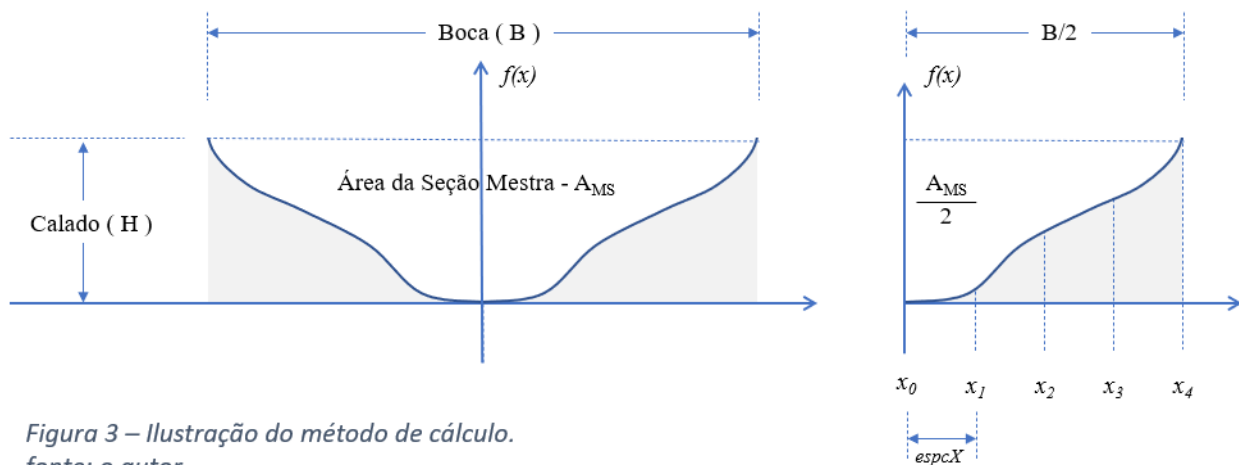


Figura 3 – Ilustração do método de cálculo.  
fonte: o autor

Considere a Figura 3. Nela, é mostrado o gráfico de uma função  $f(x)$ . Esse gráfico mostra na cor branca a área  $A_{MS}$  que deve ser calculada. Em virtude da simetria existente na figura em relação ao eixo vertical podemos trabalhar com a metade do desenho. Essa é a forma usual como os projetistas navais trabalham e é o que faremos aqui. Então considerando apenas o lado direito da seção, sabemos que: se a função  $f(x)$  for conhecida, a área cinza abaixo da curva do gráfico pode ser calculada através da integral de  $f(x)$  definida entre os pontos  $x_0$  e  $x_4$  multiplicada por 2.

A área  $A_{MS}$  será dada pela subtração: área do retângulo dado por Boca x Calado ( $B \times H$ ) menos a área cinza.

$$A_{MS} = B \times H - 2 \times \int_{x_0}^{x_4} f(x) dx$$

Sabendo isso, o problema consiste em calcular essa integral. Simples assim!!

Ocorre que existe um método numérico, denominado Fórmula de Simpson Composta, que permite que o cálculo da área cinza seja rapidamente executado em computadores caso a função  $f(x)$  seja conhecida.

Nos estágios iniciais do projeto de uma embarcação, é possível escolher algumas funções simples para um cálculo preliminar aproximado. Uma família de funções pode ser particularmente útil nesses casos: os polinômios. A título de exemplo, seja o polinômio dado pela expressão a seguir, dizemos que esse é um polinômio de grau 4, pois esse é o maior expoente aplicado à variável  $x$  e  $c_4, c_3, c_2, c_1, c_0$  são os coeficientes de cada termo.

$$f(x) = c_4 \times x^4 + c_3 \times x^3 + c_2 \times x^2 + c_1 \times x + c_0$$

Usando polinômios com coeficientes bem ajustados e a Fórmula de Simpson Composta pode-se rapidamente obter o cálculo da área  $A_{MS}$ .

## Fórmula de Simpson Composta

Este método numérico consiste em calcular a área desejada aproximando-a pela somatória das áreas de parábolas calculadas a cada três pontos adjacentes. Como se trata de uma aproximação, pode haver um erro no cálculo. Porém, escolhendo-se uma quantidade suficiente de pontos, esse erro diminui até o resultado ser bastante aceitável. Além disso, o método também permite a estimativa da ordem de grandeza do erro cometido, de tal de modo que o projetista tem informações suficientes para saber se pode ou não aplicar o método em cada situação. Maiores detalhes sobre este método numérico podem ser obtidos na literatura matemática. Aqui vamos nos concentrar na aplicação do mesmo.

Considerando o caso ilustrado na Figura 3, precisamos arbitrar a quantidade de pontos  $x_i$  a serem usados ( $QtdPtos$ ), ressaltando que sempre deve ser escolhida uma quantidade ímpar de pontos (é algo intrínseco ao método). Vamos, inicialmente, arbitrar 5 pontos conforme mostrado na figura.

Feita a escolha da quantidade de pontos devemos calcular o espaçamento  $espcX$  entre dois pontos adjacentes:

$$espcX = \frac{Boca}{2 \times (QtdPtos - 1)}$$

No nosso exemplo então teremos  $espcX = 6 / (2 \times (5 - 1)) = 0,75$

O próximo passo é escolher a função geradora. Vamos utilizar o seguinte polinômio de grau 4:

$$f(x) = 0,04321 \times x^4$$

De posse da função geradora devemos calcular  $f(x_i)$  para cada  $x_i$ . Com isso teremos os valores exibidos no quadro:

$x_0$	$x_1$	$x_2$	$x_3$	$x_4$
0,0	0,75	1,50	2,25	3,0
$f(x_0)$	$f(x_1)$	$f(x_2)$	$f(x_3)$	$f(x_4)$
0,0	0,013672	0,218751	1,107425	3,50001

e em seguida devemos aplicar os valores  $f(x)$  na seguinte expressão:

$$A_{cinza} = 2 \cdot \frac{espcX}{3} * (f(x_0) + 4 \cdot f(x_1) + 2 \cdot f(x_2) + 4 \cdot f(x_3) + f(x_4))$$

Nela, é preciso especial atenção aos coeficientes aplicados a cada termo  $f(x)$ : o primeiro e o último termos terão coeficiente 1; os demais termos de índice ímpar terão coeficiente 4; e os demais termos de índice par terão coeficiente 2. Executando as operações teremos como resultado deste exemplo:  $A_{cinza} = 4,21095m^2$

A área da seção mestra será:  $A_{MS} = B \times H - A_{cinza} = 6 \times 3,5 - 4,21095 \Rightarrow A_{MS} = 16,78905m^2$

E o coeficiente de seção mestra será:  $C_M = A_{MS}/(B \times H) = 16,78905/(6 \times 3,5) \Rightarrow C_M \approx 0,799478594...$

Vamos trabalhar o CM com 4 casas depois da vírgula, então teremos:  $C_M \approx 0,7995$

Neste mesmo caso, se aumentarmos para 7 o número de pontos arbitrados, o espaçamento  $espcX$  será reduzido para 0,5 e o cálculo se tornará mais preciso gerando um  $C_M = 0,799896548...$  e se restringirmos em 4 o número de casas decimais teremos:  $C_M \approx 0,7999$ .

Este problema é uma homenagem à Fatec de Jahu, onde, há mais de 30 anos, iniciei a minha carreira como professor.

## Entrada

A entrada consiste em um único caso de teste e é composta por 4 linhas de informação. Na primeira linha estão dois números reais positivos que representam a Boca ( $B$ ) e o Calado do navio ( $H$ ) -  $B > 0$  e  $H > 0$ .

Na segunda linha, há um número inteiro que representa o grau ( $G$ ,  $1 \leq G \leq 30$ ) do polinômio a ser usado nos cálculos. Na terceira linha estarão  $G + 1$  números reais que serão os coeficientes aplicáveis a cada termo do polinômio, separados por espaços em branco e em ordem do maior expoente para o menor.

Na quarta e última linha estará um número inteiro que representa a quantidade de pontos a ser usada nos cálculos, lembrando que esse número é sempre ímpar e  $5 \leq QtdePtos \leq 101$ .

Todas as variáveis reais usadas no programa devem ser de precisão dupla.

## Saída

A saída consiste em exibir o coeficiente de seção mestra CM com quatro casas decimais e seguido pelo fim de linha.

### Exemplo de Entrada 1

6.0 3.5	0.7995
4	
0.04321 0.0 0.0 0.0 0.0	
5	

### Exemplo de Saída 1

**Exemplo de Entrada 2**

```
6.0 3.5
4
0.04321 0.0 0.0 0.0 0.0
7
```

**Exemplo de Saída 2**

```
0.7999
```

**Exemplo de Entrada 3**

```
6.0 3.5
4
0.04321 0.0 0.0 0.0 0.0
21
```

**Exemplo de Saída 3**

```
0.8000
```

**Exemplo de Entrada 4**

```
24.8 6.0
5
8.1225e-5 -0.001 -0.001 0.01 0.5 0.0
21
```

**Exemplo de Saída 4**

```
0.6040
```

Esta página foi propositadamente deixada em branco.

## Problema E

### Comprando apartamento

*Arquivo fonte:* apartamento.{ c | cpp | java | py }

*Autor:* Prof. Me. Lucio Nunes de Lira (Fatec São Paulo)

Joãozinho está pensando em morar sozinho, porém para isso é necessário primeiro ter uma casa! Como Joãozinho não gosta de dever nada para ninguém, muito menos aos bancos, decidiu que investiria todo o dinheiro que tem em uma aplicação que mensalmente pagaria uma taxa de juros sobre o dinheiro investido e, quando for possível, usará o dinheiro obtido para realizar seu sonho e comprar um apartamento.

Como Joãozinho gosta de planejamento, pediu a sua ajuda para criar um programa em que dado o valor inicial do investimento, o valor do apartamento e a taxa de juros que será paga mensalmente pelo valor investido, exibirá uma mensagem indicando quantos meses serão necessários para que realizar a compra à vista.

Note que é possível que o valor inicial investido por Joãozinho seja igual ou até mesmo superior ao valor do apartamento e, nestes casos, ele poderá comprá-lo imediatamente!

Obs.: Sabemos que em uma situação real a inflação teria um impacto significativo no valor do apartamento, provavelmente aumentando seu valor no decorrer do tempo. Porém, para efeito de simplificação, faça um esforço de imaginação para ignorar que a inflação existe.

#### Entrada

As três linhas da entrada contêm números reais positivos que devem ser armazenados como números de ponto flutuante de dupla precisão. A primeira linha contém o valor inicial de investimento  $I$ ; a segunda linha contém o valor do apartamento  $A$  e; a terceira linha contém a taxa  $T$  que será paga mensalmente pelo valor investido, inclusive considerando juros compostos, isto é, a cada mês o valor investido considerado para o cálculo da taxa será aumentado pela taxa paga no mês anterior. Note que a taxa será informada em porcentagem.

#### Saída

A saída contém uma única linha com uma de três possíveis frases: "pode comprar agora", quando o valor investido inicialmente for suficiente para comprar o apartamento sem a necessidade de esperar o primeiro pagamento de taxa, "possível em 1 mes" quando for necessário aguardar apenas um mês para comprar o apartamento e; "possível em  $X$  meses", em que  $X$  deve ser substituído pela quantidade de meses necessários para que o valor investido seja suficiente para comprar o apartamento. Todas as frases possuem apenas letras minúsculas, sem acentuação e sem aspas.

#### Exemplo de Entrada 1

50000.00 100000.00 0.5
------------------------------

#### Exemplo de Saída 1

possível em 139 meses
-----------------------

**Exemplo de Entrada 2**

```
200000.00
210000.00
10.0
```

**Exemplo de Saída 2**

```
possivel em 1 mes
```

**Exemplo de Entrada 3**

```
450000.00
450000.00
1.0
```

**Exemplo de Saída 3**

```
pode comprar agora
```

**Exemplo de Entrada 4**

```
500000.00
350000.00
0.5
```

**Exemplo de Saída 4**

```
pode comprar agora
```



## Problema F

# Retângulo numérico

Arquivo fonte: retangulo.{ c | cpp | java | py }

Autor: Prof. Me. Lucio Nunes de Lira (Fatec São Paulo)

Seu objetivo é muito simples: dadas as coordenadas (linha, coluna) de um item específico de um retângulo numérico, exibir o valor do item.

O retângulo numérico terá uma quantidade de linhas e colunas informadas pelo usuário e seus itens serão definidos conforme as seguintes regras:

- 1º) Se o item estiver na primeira linha, seu valor é 0;
- 2º) Se o item estiver na última coluna, seu valor é 1;
- 3º) Se o item estiver na última linha, seu valor é 2;
- 4º) Se o item estiver na primeira coluna, seu valor é 3;
- 5º) Os demais itens são o resultado da soma de outros três itens: (I) o item imediatamente acima, (II) o item imediatamente à esquerda, (III) o item que está na linha acima e na coluna à esquerda.

Note que existem precedências entre as regras. Por exemplo, se o item estiver na primeira linha e na última coluna, seu valor será zero, pois a regra (1) precede a regra (2), da mesma forma que a regra (2) precede a (3) e assim por diante.

Veja a Figura 1 com uma representação de um retângulo numérico de 7 linhas por 5 colunas. Se fosse solicitado o item (4, 3) o programa deveria exibir 39, isto é, o valor do item na linha 4 e coluna 3.

0	0	0	0	0
3	3	3	3	1
3	9	15	21	1
3	15	39	75	1
3	21	75	189	1
3	27	123	387	1
2	2	2	2	1

Figura 1 – Exemplo de triângulo numérico de 7 linhas por 5 colunas.

### Entrada

A entrada contém apenas duas linhas. A primeira linha contém dois números naturais positivos  $QL$  e  $QC$  ( $0 < QL, QC \leq 25$ ) separados por um espaço, sendo  $QL$  a quantidade de linhas do retângulo e  $QC$  a quantidade de colunas. A segunda linha também contém dois números naturais positivos  $L$  ( $0 < L \leq$

$QL$  e  $C$  ( $0 < C \leq QC$ ) que representam as coordenadas (linha, coluna), respectivamente, do item do retângulo que deve ser exibido.

## Saída

O valor do item (L, C) do retângulo numérico.

### Exemplo de Entrada 1

5 6	9
3 2	

### Exemplo de Saída 1

### Exemplo de Entrada 2

7 5	39
4 3	

### Exemplo de Saída 2

### Exemplo de Entrada 3

10 12	387
6 4	

### Exemplo de Saída 3

### Exemplo de Entrada 4

25 25	59546746261147635
24 24	

### Exemplo de Saída 4

## Problema G

### Preço incorreto

Arquivo fonte: preco.{ c | cpp | java | py }

Autor: Prof. Me. Erico de Souza Veriscimo (IFSP São Miguel Paulista)

José foi almoçar no bandejão da sua faculdade que tinha acabado de ser inaugurado. Chegando lá, ao pesar seu prato, percebeu que o valor estava sendo cobrado de maneira equivocada.

Então, sabendo que você é um excelente programador, José solicitou a sua ajuda para criar um algoritmo que, dados a tara do prato, o peso e o valor do quilo, calcula de maneira correta o valor final do prato.

#### Entrada

A entrada consiste em um único caso de teste e é composta por 3 linhas. Na primeira linha, é informado um número real positivo  $Tr$  que representa a tara do prato  $Tr \geq 0$ . Na segunda linha, há um número real positivo  $P$  que representa o peso do prato  $P \geq Tr$ . E por fim, na terceira linha é informado um número real positivo  $V$  que representa o valor do quilo  $V \geq 0$ . A quantidade de casas decimais na entrada é inferior ao limite de um ponto flutuante de dupla precisão.

#### Saída

A saída consiste em exibir o valor final do prato com duas casas decimais e seguido pelo fim de linha.

##### Exemplo de Entrada 1

0.300 0.800 25.00	12.50
-------------------------	-------

##### Exemplo de Saída 1

##### Exemplo de Entrada 2

0.123 1.123 32.00	32.00
-------------------------	-------

##### Exemplo de Saída 2

Esta página foi propositadamente deixada em branco.

## Problema H

# Inimigos no Mapa

*Arquivo fonte:* inimigosnomapa.{ c | cpp | java | py }

*Autor:* Prof. Me. Sérgio Luiz Banin (Fatec São Paulo e Fatec São Caetano do Sul)

Você foi convidado por uma amiga a participar do projeto de um jogo digital. A equipe anda muito atarefada com vários aspectos do jogo e pediu para você contribuir com um algoritmo capaz de exibir uma informação específica na tela do jogo.

Trata-se de um jogo de tiro em primeira pessoa onde o objetivo principal é derrotar vários inimigos que aparecem no cenário. Na interface, é exibido um mapa em miniatura mostrando a localização aproximada dos inimigos. Esse mapa é dividido em quatro quadrantes de tamanhos iguais, designados de I, II, III e IV em números romanos maiúsculos, conforme mostrado na figura.

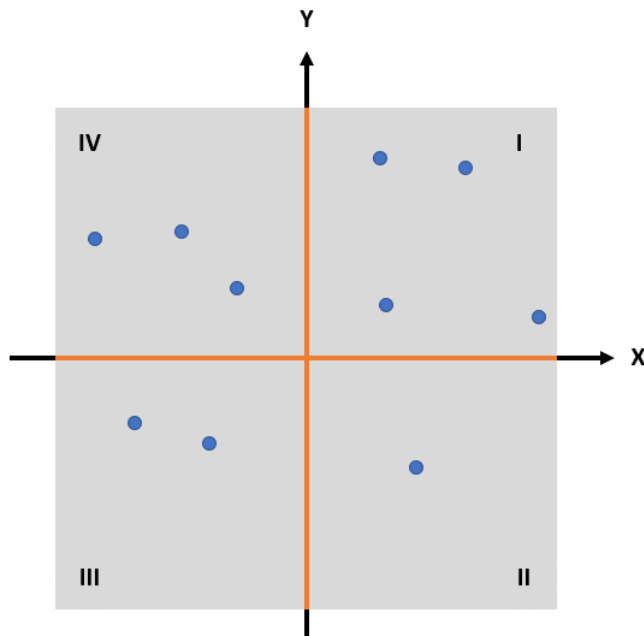


Figura 1 – Mapa e seus quadrantes

No jogo, os quadrantes são separados por muros intransponíveis, de modo que os inimigos sempre se mantêm no mesmo quadrante em que foram criados. Não há como atravessar a fronteira de um para o outro.

O seu trabalho é calcular quantos inimigos há em cada quadrante e exibi-los na tela em ordem de quadrante, primeiro o I, depois o II e assim por diante. Para isso, você sabe quantos inimigos há no mapa e também sabe as coordenadas X e Y de cada um.

A coordenada X é contada na horizontal, sendo que, na metade esquerda do mapa (quadrantes III e IV), ela é negativa e na metade direita do mapa (quadrantes I e II) ela é positiva.

A coordenada Y é contada na vertical, sendo que, na metade inferior do mapa (quadrantes II e III), ela é negativa e, na metade superior do mapa (quadrantes I e IV), ela é positiva.

Como os inimigos não podem atravessar o muro, as coordenadas X e Y jamais são iguais a zero.

## Entrada

A entrada é constituída de apenas um caso de teste. Na primeira linha haverá um número inteiro  $Q_{Inimigos}$  ( $0 < Q_{Inimigos} \leq 100$ ) que representa a quantidade de inimigos. Em seguida haverá  $Q_{Inimigos}$  linhas contendo dois números inteiros cada uma, separados por um espaço em branco, que representam as coordenadas  $X$  e  $Y$  do inimigo. É importante lembrar que tanto  $X$  como  $Y$  podem ser números positivos ou negativos, mas jamais serão zero. Saiba também que estas linhas não estão organizadas em qualquer ordem específica.

## Saída

Conforme pode ser visto nos exemplos, a saída deverá conter 4 linhas, uma para cada quadrante, informando quantos inimigos existem ali. O formato de exibição é o identificador do quadrante, na forma de número romano maiúsculo, seguido de um espaço em branco, o sinal de igual, outro espaço em branco e a quantidade de inimigos calculada. Não esqueça de acrescentar o pulo de linha no final.

### Exemplo de Entrada 1

```
10
3 6
14 21
-1 -14
-5 19
23 15
-2 3
16 9
-6 -8
19 -2
-13 6
```

### Exemplo de Saída 1

```
I = 4
II = 1
III = 2
IV = 3
```

### Exemplo de Entrada 2

```
5
1 1
1 -1
-1 1
-1 -1
2 2
```

### Exemplo de Saída 2

```
I = 2
II = 1
III = 1
IV = 1
```

## Problema I

# Contagem

*Arquivo fonte:* contagem.{ c | cpp | java | py }

*Autor:* Prof. Dr. Alex Marino (Fatec Ourinhos)

Joãozinho tem se tornado um grande filantropo e apaixonou-se pela educação fundamental. Ao retornar para sua terra natal encontrou sua professora do pré-primário e primeiros anos. Sua professora (tia Lurdes) lembrou como Joãozinho tinha dificuldade com números, mas o confortou ao dizer que ele não era o único e que muitos alunos apresentavam a mesma dificuldade. Uma dificuldade que chamou a atenção de Joãozinho é a dificuldade em contar, pois é muito comum quando as crianças aprendem a contar é que elas esquecem números. Por exemplo, no início eles podem contar “um, dois, três, cinco, seis”. Você tem que lembrá-los sobre aquele “quatro” que eles não disseram.

Na ansiedade dos pequeninos em acertar e ficarem rápidos e mostrarem proficiência na contagem, eles podem usar a maneira "rápida" de contar, por exemplo: “um, dois, pule alguns, noventa e nove, cem!” Com objetivo de ajudar as criancinhas, Joãozinho pede sua ajuda para criar um programa que ajude os pequeninos a identificar os números que faltam quando eles estiverem contando.

### Entrada

A primeira linha de entrada contém um único inteiro  $n$ , onde  $1 \leq n \leq 100$ . Cada uma das próximas  $n$  linhas contém um número que a criança recita. Cada número recitado é um número inteiro entre 1 e 200 (inclusive). Eles estão listados em ordem crescente, e não há duplicatas.

### Saída

Se a criança recitou todos os números entre 1 e o último número, imprima **bom trabalho**. Se a criança errou algum número entre 1 e o último número que recitou, imprima os números que faltam em ordem numérica crescente, um por linha.

#### Exemplo de Entrada 1

9	1
2	3
4	6
5	12
7	
8	
9	
10	
11	
13	

#### Exemplo de Saída 1

**Exemplo de Entrada 2**

5  
1  
2  
3  
4  
5

**Exemplo de Saída 2**

bom trabalho