

# CWI

## Code War

Edição 1/2 0 2 4

**Fatec**  
Ribeirão Preto

**CPS**  
Centro  
Paula Souza

## CADERNO DE QUESTÕES

**ETAPA ÚNICA**

**04 de maio de 2024**

APOIO

**CCM** 

 **S4T**

@ads.fatec Ribeirão Preto

# 1 Instruções

Este caderno contém 6 problemas – identificados por letras de A até F, com páginas numeradas de 3 até 16. Verifique se seu caderno está completo.

Informações gerais

## 1. Sobre a competição

- (a) A competição possui duração de 3 horas (início as 09h término as 12h);
- (b) É permitido a consulta a materiais já publicados anteriormente ao dia da competição
- (c) Não é permitido a comunicação com outras pessoas que não sejam a equipe para tirar dúvidas sobre a competição
- (d) É vedada a comunicação entre as equipes durante a competição, bem como a troca de material de consulta entre elas;
- (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) As linguagens permitidas são: C, C++ (extensão cc), Java e Python 3
- (g) NÃO é permitido o uso de notebooks ou outro tipo de computador ou assistente pessoal;
- (h) 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 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.

## Problema A

### Industria Mágica

Arquivo fonte: `industria.{ c | cc | java | py }`  
Autor: Lucas Baggio Figueira (FATEC Ribeirão Preto)

Uma determinada indústria dos tempos de outrora deve fazer a separação dos frascos de poção que possuem cores diferentes devido sua finalidade. Um dos gnomos operários dessa indústria foi designado para a tarefa de fazer a separação destes frascos para encaixotá-los, e, como todo o bom gnomo operário ele gosta de diferentes formas de separação para tornar seu trabalho menos entediante, sendo assim ele utiliza uma poção que separa os frascos a cada N frascos sistematicamente até que todos estejam separados. Por exemplo, caso a fila tenha 7 frascos os quais são tirados de 2 em dois até que o frasco 7 seja retirado por último, veja,

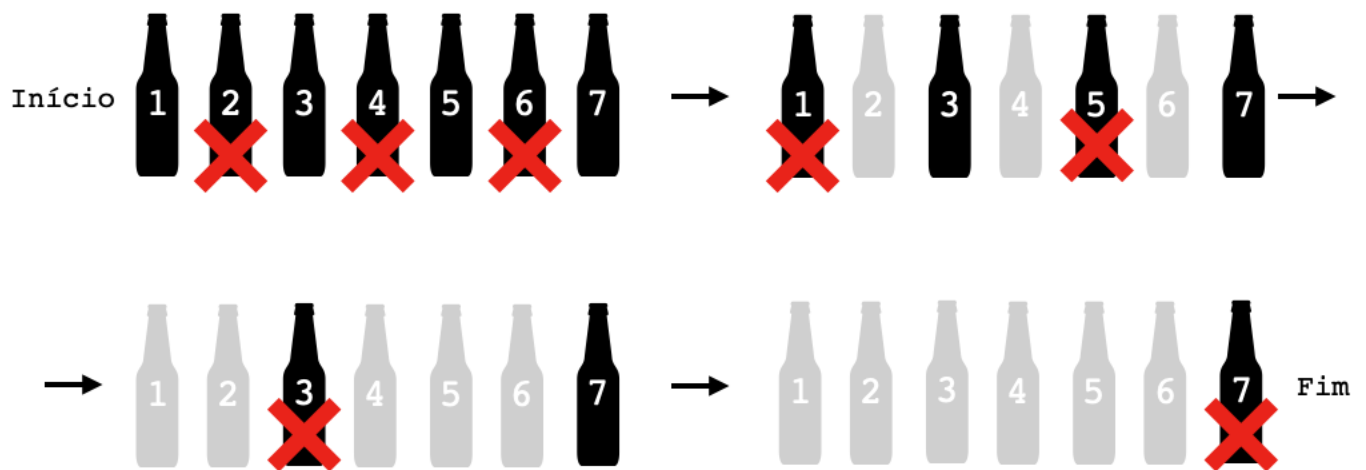


Figura A.1: Sequencia de encaixotamento dos frascos

### Entrada

A entrada contém vários casos de teste, cada caso é expresso com 2 inteiros A ( $1 < A < 50000$ ) e B ( $1 < B < 5000$ ), onde A indica a quantidade de garrafas numeradas de forma crescente a partir de 1, e B o tamanho do passo utilizado na remoção sistemática de frascos. A entrada termina com EOF.

### Saída

A saída deve, em cada linha, conter o número do último frasco para sua respectiva entrada.

#### Exemplo de Entrada 1

```
10 2
6 4
4567 123
```

#### Exemplo de Saída 1

```
5
5
60
```

Esta página foi propositadamente deixada em branco.

Problema B  
Fluxonator

Arquivo fonte: fluxonator.{ c | cc | java | py }  
Autor: Lucas Baggio Figueira (FATEC Ribeirão Preto)

Um laboratório ultrassecreto tem tentado manipular elétrons de maneira a desenvolver novas tecnologias, e, portanto, criaram uma armadilha para capturá-los de maneira uniforme, tal armadilha possui três entradas, denominadas *A*, *B*, *C*, as quais recebem elétrons de um meio diverso, e duas saídas denominadas, *D* e *E*, que direcionam tais elétrons para um meio controlado. Dentro da armadilha existem 3 alavancas (*L*<sub>1</sub>, *L*<sub>2</sub>, *L*<sub>3</sub>) que direcionam o fluxo de elétrons para alguma das saídas, sendo que cada vez um elétron passa por uma alavanca, a interação eletromagnética faz com que essa alavanca inverta sua posição, e assim, o próximo elétron que passar por ela será direcionado para outro caminho. Na figura abaixo à esquerda é possível ver o fluxo seguido pelo elétron a partir da entrada *C*, e à direita a armadilha após a mudança das alavancas. Para cada sequência de captura, a armadilha é resetada ficando com *L*<sub>1</sub>, *L*<sub>2</sub>, *L*<sub>3</sub> virados para a esquerda.

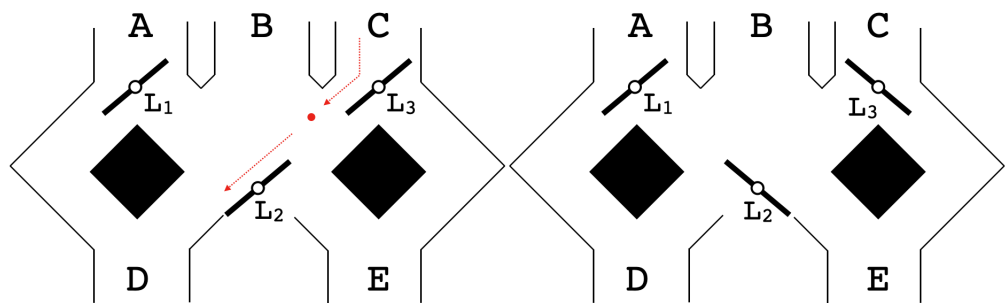


Figura B.1: Fluxonator em ação

Entrada

A entrada contém vários casos de teste, inicialmente tem-se um inteiro *N* ( $1 < N < 1000$ ) que indica quantos casos de teste serão fornecidos, cada um dos *N* casos é formado por uma string indicando a sequência de entradas por onde os elétrons são capturados.

Saída

A saída deve, em cada linha, conter uma string indicando a sequência de saída dos elétrons.

Exemplo de Entrada 1	Exemplo de Saída 1
3	DDED
ABAA	DEDDDEDED
BBBAABCCC	DEDDDEEDEDDEDEDEDEDEDED
BCABCBCAAAAACBBABACCCA	

Esta página foi propositadamente deixada em branco.

## Problema C

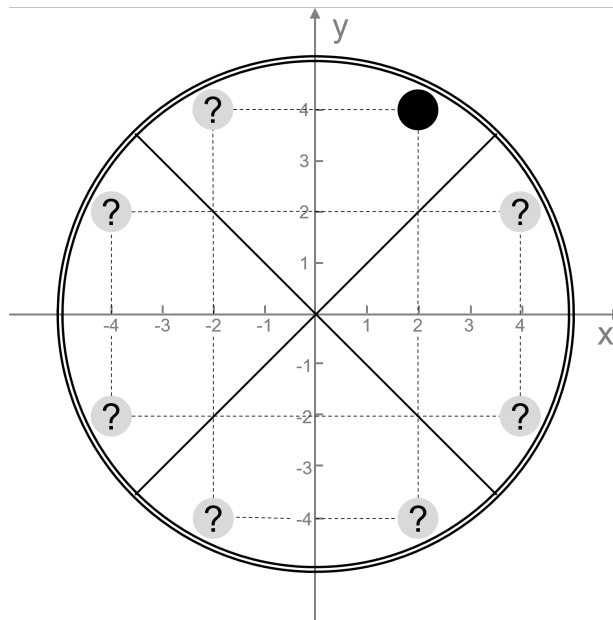
### Azeitonas na Pizza

Arquivo fonte: `azeitonas.{ c | cc | java | py }`

Autor: Rodrigo Plotze (FATEC Ribeirão Preto)

João da Silva é um apaixonado pela Matemática e sempre busca oportunidades de utilizá-la nas mais variadas situações do cotidiano. Um dia desses em sua casa fez o pedido de uma pizza por um aplicativo de delivery, e quando recebeu a pizza ficou pensando no seguinte problema: Existe algum método para determinar simetricamente o posicionamento de oito azeitonas na área do círculo da pizza?

Para ilustrar este problema, observe a imagem:



Na imagem, uma azeitona foi posicionada no *plano cartesiano da pizza* na coordenada  $(2, 4)$ . Diante disso, como é possível determinar o posicionamento das sete azeitonas restantes de forma a obter uma distribuição simétrica?

#### Entrada

A entrada começa com uma linha com dois inteiros:  $N$  ( $1 \leq N \leq 100$ ) representando a coordenada  $(x, y)$  da posição de uma azeitona no *plano cartesiano da pizza*. A coordenada é válida se, e somente se,  $x \neq y$ . A última linha de entrada termina com uma quebra de linha.

#### Saída

Como saída apresente as oito coordenadas, uma coordenada em cada linha, simétricas que representam o posicionamento das azeitonas na pizza. As linhas da saída seguem as posições das azeitonas no sentido

horário. Caso a entrada seja inválida, apresente a mensagem *ERRO*. A última linha de saída termina com uma quebra de linha.

**Exemplo de Entrada 1**

2 4

**Exemplo de Saída 1**

2 4  
4 2  
4 -2  
2 -4  
-2 -4  
-4 -2  
-4 2  
-2 4

**Exemplo de Entrada 2**

-4 -3

**Exemplo de Saída 2**

ERRO

**Exemplo de Entrada 3**

5 1

**Exemplo de Saída 3**

5 1  
1 5  
1 -5  
5 -1  
-5 -1  
-1 -5  
-1 5  
-5 1

**Exemplo de Entrada 4**

2 9

**Exemplo de Saída 4**

2 9  
9 2  
9 -2  
2 -9  
-2 -9  
-9 -2  
-9 2  
-2 9

**Exemplo de Entrada 5**

0 2

**Exemplo de Saída 5**

ERRO



## Problema D

### Estacionamento do Seu Zé

Arquivo fonte: estacionamento.{ c | cc | java | py }

Autor: Rodrigo Plotze (FATEC Ribeirão Preto)

O estacionamento do *Seu Zé* fica localizado no município brasileiro de Passa e Fica, no interior do estado do Rio Grande do Norte. Neste município todas as bicicletas são obrigatoriamente identificadas com placas. Uma placa é definida por um conjunto de 7 (sete) caracteres alfanuméricos, com combinação aleatória de 4 (quatro) letras e 3 (três) números. O estacionamento possui, atualmente, um total de 15 (quinze) vagas identificadas numericamente.



Recentemente o *Seu Zé* implantou no estacionamento um sistema que determina automaticamente a vaga em que a bicicleta deve ser estacionada. O sistema realiza o reconhecimento óptico de caracteres da placa e determina a posição da bicicleta no estacionamento a partir das letras e dos números. Por exemplo, a placa *ABC1D23* deverá ser estacionada na vaga 12. Para isso, o seguinte cálculo foi realizado:

$$P = 65 + 66 + 67 + 1 + 68 + 2 + 3 = (272\%T) + 1 = 12$$

Em que, os valores 65, 66, 67 e 68 representam os valores decimais dos caracteres *A, B, C* e *D* na Tabela ASCII e *T* o número total de vagas do estacionamento. Quando uma vaga está ocupada não é possível estacionar outra bicicleta, dessa forma, o dono da bicicleta precisará procurar outro estacionamento

#### Entrada

Uma lista de placas de bicicletas que desejam uma vaga no estacionamento do *Seu Zé*. A última linha de entrada termina com uma quebra de linha.

#### Saída

A saída deve, em cada linha, apresentar as placas de bicicletas que conseguiram uma vaga no estacionamento. A listagem deve conter o número da vaga e a placa da bicicleta. As demais placas deverão ser ignoradas. A última linha de entrada termina com uma quebra de linha.

**Exemplo de Entrada 1**

ABC1D23  
QNT8B49  
JBO5T18  
GDK2W13  
GXA4D66  
RRP4T27  
ACP9A44  
SLS7B62  
GRO2F24  
EQY8F35  
QGI1Y43  
XKN8V47  
IJT5M37  
TYE7K36  
DZH8I89  
QIQ3G43  
CDO5S18  
MUZ2Y29  
YEQ4E35  
RLG7H29

**Exemplo de Saída 1**

2 GDK2W13  
3 GXA4D66  
4 ACP9A44  
5 GRO2F24  
6 RRP4T27  
8 DZH8I89  
10 QNT8B49  
11 XKN8V47  
12 ABC1D23  
13 TYE7K36  
14 RLG7H29  
15 YEQ4E35

## Problema E

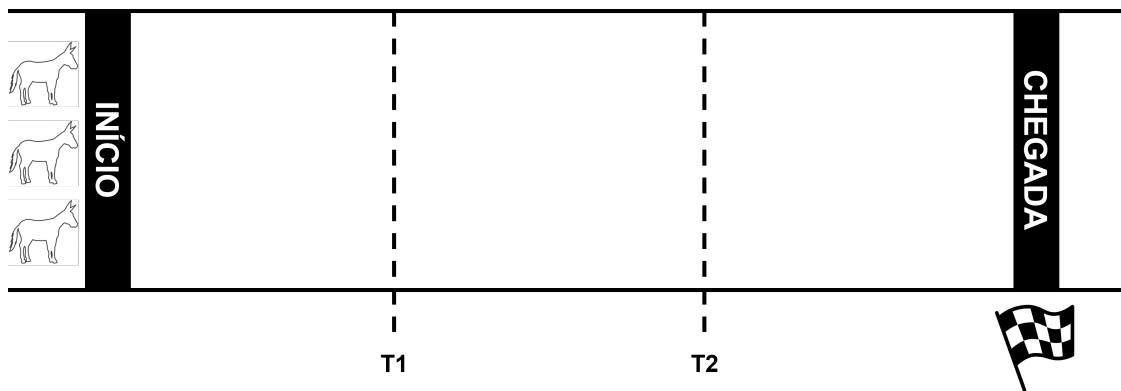
### De quem é esse Jegue?

*Arquivo fonte:* jegues.{ c | cc | java | py }

*Autor:* Rodrigo Plotze (FATEC Ribeirão Preto)

A Corrida de Jegue é um evento que acontece anualmente na cidade de Itabi, localizada no interior do estado de Sergipe. Este evento atrai milhares de pessoas das mais diversas partes do mundo. Qualquer pessoa pode participar da competição, desde que, é claro, seja capaz de controlar o seu jegue através do percurso de 300 metros rua abaixo.

A cada ano a competição conta com um número maior de inscritos e com isso tem aumentado a dificuldade para determinar os três primeiros colocados da prova. Para resolver este problema a equipe de organizadores pensa em adicionar dispositivos de telemetria nos animais, e assim, realizar a coleta de informações precisas durante a realização da prova. O percurso contará com três pontos de coleta de dados, indicados na como *T1*, *T2* e *CHEGADA*. Em cada ponto de coleta é realizada leitura do tempo de cada participante em milisegundos.



A equipe de organizadores deseja saber quais os três primeiros colocados da competição em cada ponto de coleta. Assim, você deverá escrever uma solução computacional capaz de apresentar os nomes dos três primeiros colocados no T1, T2 e CHEGADA.

### Entrada

A entrada é composta por uma lista contendo em cada linha o nome do competidor, o tempo em milisegundos no ponto de coleta T1, o tempo em milisegundos no ponto de coleta T2 e o tempo em milisegundos na linha de chegada. Os dados, em cada linha, são separados por um espaço em branco.

### Saída

A saída deve apresentar os nomes dos três primeiros colocados em cada ponto de coleta. Para cada ponto de coleta deve ser apresentado, em uma única linha, o nome do ponto de coleta (T1, T2 ou CHEGADA), o nome do primeiro colocado, o nome do segundo colocado e o nome do terceiro colocado. Por exemplo:

T1 João José Maria

T2 Ana João José  
 CHEGADA João Maria José

### Exemplo de Entrada 1

Willa 65877 128839 207488  
 Hayley 65287 124817 193510  
 Ina 60175 122192 198273  
 Ezekiel 78636 121501 198047  
 Desirae 69080 134192 216968  
 Xandra 62580 136605 198388  
 Gillian 75148 134331 199639  
 Pascale 71783 130409 192810  
 Fay 68518 121211 191793  
 Roth 65954 131476 214952

### Exemplo de Saída 1

T1 Ina Xandra Hayley  
 T2 Fay Ezekiel Ina  
 CHEGADA Fay Pascale Hayley

### Exemplo de Entrada 2

Jamal 77261 133536 202025  
 Nadine 72472 122640 208786  
 Vance 74614 122907 198825  
 Shad 72467 127607 217506  
 Paul 79513 135707 194897  
 Lyle 72265 122130 191725  
 Stephen 79377 130456 209896  
 Benjamin 62816 139920 191728  
 Paul 65393 129809 188888  
 Mona 75037 133851 188172  
 Rashad 74320 121099 205251  
 Thane 76073 134913 182169  
 Clio 66988 126774 209863  
 Kermit 60082 128677 198205  
 Gareth 65945 133625 191706  
 Lance 75956 126494 218753  
 Dorian 67009 128701 204336  
 Jocelyn 69166 132302 196753  
 Graham 76864 125035 215506  
 Raja 66841 133749 181153

### Exemplo de Saída 2

T1 Kermit Benjamin Paul  
 T2 Rashad Lyle Nadine  
 CHEGADA Raja Thane Mona

**Exemplo de Entrada 3**

```
Colin 64785 135393 194035
Laurel 68326 129286 198369
Linus 79888 131039 205647
Laith 67533 133213 193860
Jin 61684 132355 185737
Zoe 68419 135238 185823
Ella 76993 120402 201008
Castor 72521 125650 219183
Nyssa 63934 138254 186451
Montana 74544 134276 204389
Daryl 62311 136295 184134
Simon 76147 129104 201212
Lewis 66968 138090 210673
Axel 62789 128316 215822
Kenneth 77535 127659 180061
```

**Exemplo de Saída 3**

```
T1 Jin Daryl Axel
T2 Ella Castor Kenneth
CHEGADA Kenneth Daryl Jin
```

Esta página foi propositadamente deixada em branco.

## Problema F

### Batatinha Frita 1,2,3

*Arquivo fonte:* batatinha.{ c | cc | java | py }

*Autor:* Lucas Baggio Figueira (FATEC Ribeirão Preto)

Um grupo aficionado por jogos extremos criou um desafio de vida ou morte, onde o competidor tem, no momento que ouvir Batatinha Frita 1,2,3, que encontrar o caminho mais curto por meio de um complicado labirinto. Assim que o competidor em questão chegar à saída do labirinto ele deverá acionar uma alavanca que impede que uma quantidade potencialmente letal de gás mostarda seja despejada no ambiente em questão. Portanto, os competidores que tenham a coragem de se submeter à este desafio devem usar toda a sua intuição, agilidade e capacidade física.

#### Entrada

A entrada contém vários casos de teste, configurada da seguinte maneira, na primeira linha tem-se um inteiro  $N$  ( $1 < N < 100$ ) que indica o tamanho do labirinto ( $N \times N$ ), na linha subsequente tem-se o ponto de entrada formado por  $a$  ( $0 < a \leq N$ ) e  $b$  ( $0 < b \leq N$ ) representando, respectivamente, linha e coluna onde o competidor deverá entrar no labirinto, logo abaixo tem-se  $c$  ( $0 < c \leq N$ ) e  $d$  ( $0 < d \leq N$ ), representando a linha e coluna, respectivamente, de onde há a alavanca de segurança. Por fim, tem-se o labirinto ( $N \times N$ ) onde cada posição pode conter 0 ou 1, sendo que 1 indica uma passagem válida no labirinto.

#### Entrada

A saída deve conter a sequência do caminho mais curto entre  $(a,b)$  e  $(c,d)$ .

**Exemplo de Entrada 1**

```
10
10 6
2 10
0000000000
1100011111
0110010000
0011110000
0010011110
1110000010
0011110000
0000011111
0000010000
0000010000
```

**Exemplo de Saída 1**

```
10 6
9 6
8 6
7 6
7 5
7 4
7 3
6 3
5 3
4 3
4 4
4 5
4 6
3 6
2 6
2 7
2 8
2 9
2 10
```

**Exemplo de Entrada 2**

```
5
1 3
5 5
00100
00111
10101
11101
00001
```

**Exemplo de Saída 2**

```
1 3
2 3
2 4
2 5
3 5
4 5
5 5
```