



## Lista de Exercícios 9

1. (Maratona de Programação da SBC – ACM ICPC – 2013) Zerinho ou um.

Todos devem conhecer o jogo Zerinho ou Um (em algumas regiões também conhecido como Dois ou Um), utilizado para determinar um ganhador entre três ou mais jogadores. Para quem não conhece, o jogo funciona da seguinte maneira. Cada jogador escolhe um valor entre zero ou um; a um comando (geralmente um dos competidores anuncia em voz alta “Zerinho ou... Um!”), todos os participantes mostram o valor escolhido, utilizando uma das mãos: se o valor escolhido foi um, o competidor mostra o dedo indicador estendido; se o valor escolhido foi zero, mostra a mão com todos os dedos fechados. O ganhador é aquele que tiver escolhido um valor diferente de todos os outros; se não há um jogador com valor diferente de todos os outros (por exemplo todos os jogadores escolhem zero, ou um grupo de jogadores escolhe zero e outro grupo escolhe um), não há ganhador.

Alice, Beto e Clara são grandes amigos e jogam Zerinho a toda hora: para determinar quem vai comprar a pipoca durante a sessão de cinema, quem vai entrar na piscina primeiro, etc. Jogam tanto que resolveram fazer um plugin no Facebook para jogar Zerinho. Como não sabem programar, dividiram as tarefas entre amigos que sabem, inclusive você.

Dados os três valores escolhidos por Alice, Beto e Clara, cada valor zero ou um, escreva um programa que determina se há um ganhador, e nesse caso determina quem é o ganhador.

### **Entrada**

A entrada é composta de uma única linha, que contém três inteiros A, B e C, indicando respectivamente os valores escolhidos por Alice, Beto e Clara.

### **Saída**

Seu programa deve produzir uma única linha, contendo um único caractere. Se o vencedor é Alice o caractere deve ser ‘A’, se o vencedor é Beto o caractere deve ser ‘B’, se o vencedor é Clara o caractere deve ser ‘C’ e se não há vencedor o caractere deve ser ‘\*’ (asterisco).

### **Restrições**

A, B, C  $\in \{0, 1\}$

### **Exemplo:**

Entrada:

1      1      0

Saída:

C

2. (Maratona de Programação da SBC – ACM ICPC – 2017) Despojados:

Todo inteiro positivo pode ser escrito como um produto de potências de primos. Por exemplo,  $252 = 2^2 \times 3^2 \times 7$ . Um inteiro é despojado se pode ser escrito como um produto de dois ou mais primos distintos, sem repetição. Por exemplo,  $6 = 2 \times 3$  e  $14 = 2 \times 7$  são despojados, mas  $28 = 2^2 \times 7$ , 1, 17 não são despojados.

**Entrada**

A entrada consiste de uma única linha que contém um inteiro  $N$  ( $1 \leq N \leq 1012$ ).

**Saída**

Seu programa deve produzir uma única linha com um inteiro representando o número de divisores despojados de  $N$ .

**Exemplo:**

Entrada:

252

Saída:

4

3. (Maratona de Programação da SBC – ACM ICPC – 2013) Mergulho:

O recente terremoto em Nlogônia não chegou a afetar muito as edificações da capital, principal epicentro do abalo. Mas os cientistas detectaram que o principal dique de contenção teve um dano significativo na sua parte subterrânea que, se não for consertado rapidamente, pode causar o seu desmoronamento, com a consequente inundação de toda a capital.

O conserto deve ser feito por mergulhadores, a uma grande profundidade, em condições extremamente difíceis e perigosas. Mas como é a sobrevivência da própria cidade que está em jogo, seus moradores acudiram em grande número como voluntários para essa perigosa missão.

Como é tradicional em missões perigosas, cada mergulhador recebeu no início do mergulho uma pequena placa com um número de identificação. Ao terminar o mergulho, os voluntários devolviam a placa de identificação, colocando-a em um repositório.

O dique voltou a ser seguro, mas aparentemente alguns voluntários não voltaram do mergulho. Você foi contratado para a penosa tarefa de, dadas as placas colocadas no repositório, determinar quais voluntários perderam a vida salvando a cidade.

**Entrada**

A entrada é composta de duas linhas. A primeira linha contém dois inteiros  $N$  e  $R$ , indicando respectivamente o número de voluntários que mergulhou e o número de voluntários que retornou do mergulho. Os voluntários são identificados por números de 1 a  $N$ . A segunda linha da entrada contém  $R$  inteiros, indicando os voluntários que retornaram do mergulho (ao menos um voluntário retorna do mergulho).

**Saída**

Seu programa deve produzir uma única linha, contendo os identificadores dos voluntários que não retornaram do mergulho, na ordem crescente de suas identificações. Deixe um espaço em branco após cada identificador (note que isto significa que deve haver um espaço em branco também após o último identificador). Se todos os voluntários retornaram do mergulho, imprima apenas o caractere '\*' (asterisco).

**Restrições**

$$1 \leq R \leq N \leq 104$$

**Exemplo:**

Entrada:

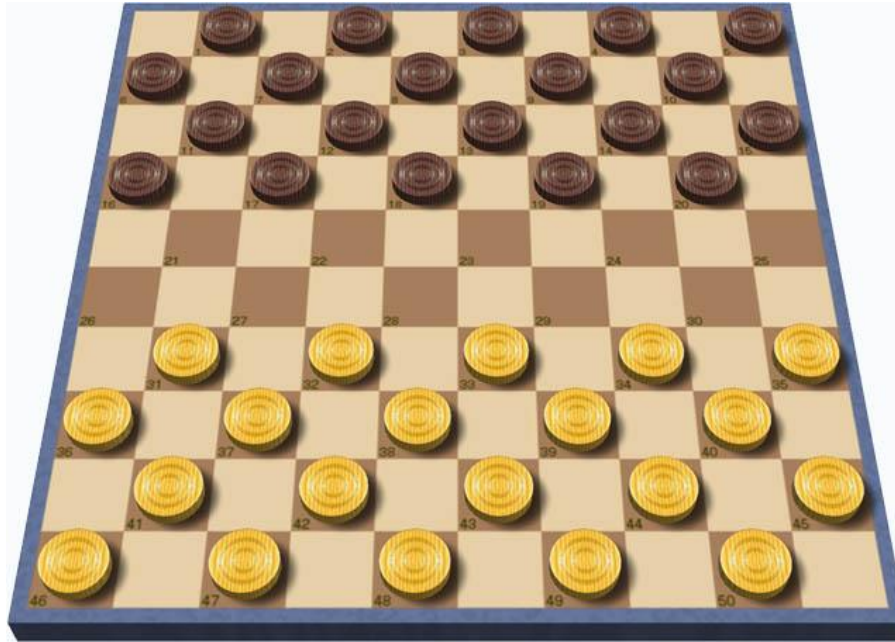
```
5      3
3      1      5
```

Saída:

```
2      4
```

#### 4. (Desafio) Damas:

O Jogo de Damas é realizado em um tabuleiro que contém 10 linhas e 10 colunas (tabuleiro internacional), como demonstrado na imagem abaixo, elaborado para dois jogadores com 20 peças cada:



Faça um algoritmo em C que permita que duas pessoas joguem Damas por meio de um programa. Lembre-se que as peças se movimentam apenas na diagonal, e uma casa de cada vez, mas a peça que alcança a décima casa é promovida a “dama”, o que lhe permite realizar movimentos maiores.

##### **Entrada:**

O tabuleiro já deve estar inicializado com as peças dos jogadores. Considere 1 como representante das peças do jogador 1, 2 como representante das peças do jogador 2 e 0 como sendo os espaços vazios. O jogo deve permitir que os jogadores façam suas jogadas indicando a linha e a coluna onde se encontra a peça a ser movimentada, e a linha e a coluna de destino, ou seja, novo local para onde a peça irá.

##### **Saída:**

A cada novo movimento deve ser apresentado o tabuleiro atualizado. Ao final do jogo, deve-se apresentar o jogador vencedor, a quantidade de peças que restaram com o jogador vencedor e quantos movimentos foram necessários para a vitória.

**Exemplo:**

Entrada:

1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	2	0	2	0	2	0	2	0	2
2	0	2	0	2	0	2	0	2	0
0	2	0	2	0	2	0	2	0	2
2	0	2	0	2	0	2	0	2	0

Jogador 1:

4      2

5      1

Saída:

1	0	1	0	1	0	1	0	1	0
0	1	0	1	0	1	0	1	0	1
1	0	1	0	1	0	1	0	1	0
0	0	0	1	0	1	0	1	0	1
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	2	0	2	0	2	0	2	0	2
2	0	2	0	2	0	2	0	2	0
0	2	0	2	0	2	0	2	0	2
2	0	2	0	2	0	2	0	2	0