



# OPEI 2016

## Olimpíada Pernambucana de Informática

### MODALIDADE C

### PROVA PRÁTICA - ENSINO MÉDIO

Leia atentamente as seguintes Instruções.

- Esta prova tem início às 09:00 (nove) horas do dia 03 de Setembro de 2016.
- Esta prova, modalidade prática, possui duração de 3 (três) horas.
- O aluno só poderá deixar o local de prova a partir de 30 (trinta) minutos do horário inicial da prova.
- Os últimos 3 alunos restantes na sala, devem esperar até o final do tempo de prova para sair.
- Aguarde orientações quanto a submissão das respostas.
- Preencha à caneta os seus dados pessoais.
- É proibido realizar qualquer tipo de pesquisa ou consulta.

Nome completo: \_\_\_\_\_

Idade: \_\_\_\_ Ano escolar do aluno: \_\_\_\_ Nº do RG ou CPF: \_\_\_\_\_

Escola: \_\_\_\_\_

Organização



Apoio



# Problema A

Luma é professora e se confunde muito na hora de atribuir a nota a seus alunos e contabilizar quantos alunos tiveram uma nota acima da média. A média na escola onde Luma trabalha é definida como a média aritmética das notas dos alunos, portanto, tirando uma nota maior ou igual a essa média, o aluno estará aprovado.

Dado um conjunto de notas dos **N** alunos de Luma, imprima o percentual de alunos aprovados. Use variáveis de ponto flutuante e imprima a resposta com duas casas decimais de precisão.

## Entrada:

A primeira linha de entrada possui o inteiro **N**, mencionado acima. Na segunda linha os **N** valores inteiros  $a_i$  separados por espaços representam a sequência de notas dos alunos de Luma.

## Saída:

Imprima o valor real que representa o percentual [0.00 - 100.00] de aprovação. Não esqueça de imprimir o símbolo de percentual '%' em seguida e a quebra de linha.

## Limites:

$$0 \leq N \leq 10^6 \quad \forall i \ 0 \leq a_i \leq 10$$

## Exemplos:

| Entrada:               | Saída:  |
|------------------------|---------|
| 5<br>1 2 3 4 5         | 60.00%  |
| Entrada:               | Saída:  |
| 6<br>1 1 1 1 1 1       | 100.00% |
| Entrada:               | Saída:  |
| 7<br>1 3 10 5 10 10 10 | 57.14%  |

*Autor: Ruy Brito*

## Problema B

Luma ama *tetraedros*, que são poliedros regulares de 4 lados (triangulares) porém não possui conhecimentos matemáticos suficientes para calcular suas áreas, volumes ou alturas. Por isso, ela pede sua ajuda!

Faça um programa que recebe um inteiro **L**, que representa o lado do tetraedro regular, e informa a altura **H**, a área total **A** e o volume **V** do tetraedro de lado **L**.

### Entrada:

A primeira e única linha de entrada possui um inteiro **L**.

### Saída:

Imprima uma única linha com os três valores reais **H**, **A** e **L** (imprima os três valores com pelo menos 2 casas decimais de precisão) espaçados por um único espaço em branco. Lembre-se da quebra de linha ao final.

### Limites:

$$0 \leq L \leq 10^3$$

### Exemplos:

|          |                               |
|----------|-------------------------------|
| Entrada: | Saída:                        |
| 7        | 5.72 84.87 40.42              |
| Entrada: | Saída:                        |
| 765      | 624.62 1013639.43 52761612.17 |

Autor: Ruy Brito

## Problema C

Vinido adora permutações de sequência numéricas. A permutação de um número **N** é definida como uma sequência de todos os números de 1 a **N**, em alguma ordem. Vinido possui vários desses números e gostaria de saber se o conjunto de números que ele tem formam uma permutação.

Ainda, Vinido se interessa pela ordem dos números. Dois tipos de permutações notáveis são as crescentes e decrescente, por exemplo a sequência {1, 2, 3, 4, 5} é uma permutação crescente de 5, enquanto que a sequência {4, 3, 2, 1} é uma permutação decrescente de 4 e a sequência {1, 3, 2} é uma permutação comum de 3.

### Entrada:

A primeira linha de entrada possui o inteiro **N**, mencionado acima. Na segunda linha os **N** valores inteiros  $a_i$  separados por espaços representam a sequência a ser analisada.

### Saída:

A única linha de saída deve conter **PERMUTACAO CRESCENTE** se a sequência dada for uma permutação crescente de N, **PERMUTACAO DECRESCENTE** se for permutação decrescente de N, **PERMUTACAO COMUM** se for uma permutação comum de N ou **ERRO**, se não for uma permutação de N. Atenção: não imprima acentuação.

### Limites:

$$0 \leq N \leq 10^3 \quad \forall i \ 1 \leq a_i \leq N$$

### Exemplos:

| Entrada:           | Saída:               |
|--------------------|----------------------|
| 5<br>1 2 3 4 5     | PERMUTACAO CRESCENTE |
| Entrada:           | Saída:               |
| 7<br>1 7 3 2 4 6 5 | PERMUTACAO COMUM     |
| Entrada:           | Saída:               |
| 5<br>1 2 2 3 4     | ERRO                 |

Autor: Ruy Brito

## Problema D

Na Roma Antiga, um dos tipos mais comuns de criptografia era a Cifra de César. Uma de suas variações, chamada ROT13, consistia de "empurrar" todas as letras 13 posições, e substituí-las na mensagem. Por exemplo, a A se torna a letra N, B se torna O, C se torna P, e por aí vai. O reverso também vale, por exemplo, Y se torna L, e Z vira M.

Entretanto, o que pouco se sabe é que não muito distante da Roma Antiga existia o império Maratônicus. Lá, o imperador Ruyus Divinus tomou como desafio pessoal criar uma cifra mais poderosa que a ROT13 de César. Ele então criou a ROT N, que consiste do mesmo mecanismo da ROT13, mas com a quantidade N de "empurrões" a ser escolhida pelo usuário na hora de escrever a mensagem.

O problema dessa técnica residia em que aqueles que recebiam a mensagem não conheciam o valor de N e não conseguiam compreendê-la. Não é à toa que ninguém nunca ouviu falar do império Maratônicus! Entretanto, você é apenas um simples mensageiro, e sua única missão é criar a mensagem em ROT N, então mãos à obra!

### Entrada:

A primeira linha da entrada consiste de um valor N que equivale à quantidade de "empurrões" em cada letra. A segunda linha consiste de uma frase que deve ser convertida para ROT N. A frase pode conter espaços e letras maiúsculas e minúsculas, mas não caracteres especiais ou acentos.

### Saída:

A saída consiste de uma única linha contendo a mensagem convertida, preservando maiúsculas, minúsculas, e espaços.

### Limites:

$$0 \leq N \leq 10^5$$

### Exemplos:

| Entrada:               | Saída:             |
|------------------------|--------------------|
| 0<br>OPEI              | OPEI               |
| Entrada:               | Saída:             |
| 5<br>Salada Cesar      | Xfqfif Hjxfw       |
| Entrada:               | Saída:             |
| 2<br>Pssinho do romano | Rcuukpqj fq tqocpq |

Autor: Lucas Valença

## Problema E

Dilu possui um tabuleiro de xadrez inicialmente todo preenchido com peões, isto é, todas as casas do tabuleiro possuem um peão posicionado sobre ela. Dilu gosta de remover as peças em retângulos da seguinte forma: a cada iteração ele escolhe quatro valores  $a$ ,  $b$ ,  $c$  e  $d$ , que representam, respectivamente, a linha inicial, a coluna inicial, a linha final e a coluna final. Isso significa que toda casa do tabuleiro  $(x, y)$  tal que  $x \in [a, c]$  e  $y \in [b, d]$ , terá seu peão removido.

Dilu gostaria de saber qual o percentual de peões que foi removido ao longo das

### Entrada:

A primeira linha de entrada possui dois inteiros **N** e **M** que representam as dimensões do tabuleiro, ou seja, indica que o tabuleiro possui **N** linhas e **M** colunas. A segunda linha de entrada possui um único inteiro **K** que representa o número de iterações (remoções) que Dilu realiza. As próximas **K** linhas possuem quatro valores  $a$ ,  $b$ ,  $c$  e  $d$  que representam as coordenadas da  $i$ -ésima iteração.

### Saída:

Imprima o valor real que representa o percentual [0.00 - 100.00] de remoção. Não esqueça de imprimir o símbolo de percentual '%' em seguida e a quebra de linha.

### Limites:

$$\begin{aligned}1 &\leq N, M \leq 5 * 10^2 \\1 &\leq K \leq 10 \\1 &\leq a \leq c \leq N \\1 &\leq b \leq d \leq M\end{aligned}$$

### Exemplos:

| Entrada:                       | Saída: | Entrada:                                  | Saída: |
|--------------------------------|--------|---|--------|
| 3 3<br>2<br>1 1 1 2<br>2 3 3 3 | 44.44% | 6 6<br>3<br>1 1 2 5<br>1 1 6 2<br>5 5 6 6 | 61.11% |

Autor: Ruy Brito

## Problema F

Luma e Vinido gostam de brincar com números. No entanto, Luma prefere os números menores, enquanto que Vinido prefere os maiores.

Eles recebem um número com alguns caracteres apagados e Luma sugere que seria melhor substituí-los de forma a obter o maior número possível, mas Vinido discorda. Ele acha que o ideal seria substituí-los de forma a obter o menor número possível (evitando os 0s à esquerda). Por exemplo, se eles recebessem o número “3?2?”, Luma formaria o número “3020”, enquanto que Vinido formaria “3929”. Qual a diferença entre os números obtidos por Luma e Vinido?

### Entrada:

A primeira e única linha de entrada possui a string **S** que representa o número com 0 ou mais caracteres ilegíveis.

### Saída:

Imprima uma única linha com a diferença entre os números formados por Luma e por Vinido.

### Limites:

$$0 \leq |S| \leq 18$$

### Exemplos:

|          |        |
|----------|--------|
| Entrada: | Saída: |
| 17834    | 0      |
| Entrada: | Saída: |
| 3?2?     | 909    |
| Entrada: | Saída: |
| ?99?     | 8009   |

*Autor: Ruy Brito*

## Problema G

Anabela gosta de contar estrelas cadentes, e, para isso, possui um calendário com  $N$  dias, numerados de 1 a  $N$ , que serve para contar quantas estrelas caíram em cada dia. Sempre que ela descobre que algum amigo viu uma estrela cadente no dia  $X$ , ela adiciona essa estrela à quantidade de estrelas vistas no dia. Às vezes ela quer saber, dado um intervalo  $[L, R]$  (do dia  $L$  até o dia  $R$ ), quantas estrelas cadentes foram vistas no total.

Porém, tem se tornado difícil de contar as estrelas na mão, e, para isso, ela quer que você faça um programa que seja capaz de ler o estado atual do seu calendário, e processar dois tipos de operações:

1. Adicionar 1 à quantidade de estrelas vistas no dia  $X$
2. Contar quantas estrelas foram vistas no total entre os dias  $L$  e  $R$ .

### Entrada:

A primeira linha é composta por um inteiro  $N$ , a quantidade de dias do calendário e um inteiro  $Q$ , a quantidade de operações que seu programa deve processar. A segunda linha é composta por  $N$  inteiros, cada  $a_i$  inteiro representando a quantidade de estrelas cadentes no dia  $i$  do calendário. As próximas  $Q$  linhas de entrada podem ser de dois tipos: “1  $X$ ” ou “2  $L R$ ”, indicando que seu programa deve realizar a operação 1 ou 2.

### Saída:

Para cada operação do tipo 2 lida na entrada, imprima uma linha contendo um único inteiro  $Y$ , a soma da quantidade de estrelas cadentes presentes no calendário de anabela entre os dias  $L$  e  $R$ .

### Limites:

$$1 \leq N, Q, X, L, R \leq 10^3 \text{ e } L \leq R$$

### Exemplos:

| Entrada:  | Saída: |
|-----------|--------|
| 5 4       | 9      |
| 1 2 1 4 1 | 1      |
| 2 1 5     | 2      |
| 2 3 3     | 10     |
| 1 3       |        |
| 2 3 3     |        |
| 2 1 5     |        |



| Entrada:            | Saída: |
|---------------------|--------|
| 10 9                | 10     |
| 0 0 0 2 5 3 0 2 3 4 | 11     |
| 2 4 6               | 14     |
| 1 1                 |        |
| 2 1 6               |        |
| 1 8                 |        |
| 1 9                 |        |
| 1 10                |        |
| 1 8                 |        |
| 1 8                 |        |
| 2 8 10              |        |
| Entrada:            | Saída: |
| 1 4                 | 8      |
| 5                   |        |
| 1 1                 |        |
| 1 1                 |        |
| 1 1                 |        |
| 2 1 1               |        |

*Autor: Higor Cavalcanti*

# Problema H

Um *palíndromo* é uma palavra que é lida da mesma maneira tanto da esquerda para a direita, quanto da direita para a esquerda. Por exemplo: “ARARA”, “REVIVER”, “ANA” e “MAAM”, são todos palíndromos, enquanto “TOTEM”, “ANTIGO” e “NANA” não.

## Entrada:

A entrada é constituída de uma única string **S**, constituída apenas por letras maiúsculas, sem acentuação.

## Saída:

A saída deve conter uma letra, “S” (sem aspas) , caso *S* seja um palíndromo ou “N” (sem aspas), caso contrário.

## Limites:

$$1 \leq |S| \leq 10^6$$

## Exemplos:

|          |        |
|----------|--------|
| Entrada: | Saída: |
| XOXOX    | S      |
| Entrada: | Saída: |
| ETEETE   | S      |
| Entrada: | Saída: |
| TOYOTO   | N      |

*Autor: Higor Cavalcanti*

# Problema I

A loja de bolsas e sapatos do Sr. Michiles recebe muitas clientes todos os dias. Por causa disso, ele acaba sobrecarregado e estressado tentando lidar com toda a demanda sozinho, sem deixar de dar atenção à cada uma de suas clientes de forma justa. Cabe a você ajudá-lo a administrar seu tempo!

As clientes podem querer fazer compras ou reparo de produtos. Cada produto comprado leva 1 unidade de tempo. Reparos levam 2 vezes mais unidades de tempo por produto do que compras.

Você deve fazer por onde o atendimento ser o mais justo possível, para que todas tenham um tempo de espera similar, dando prioridade em caso de empate a quem tiver o menor número de cliente! Lembre-se que enquanto uma está sendo atendida, as outras estão esperando!

## Entrada:

A primeira linha da entrada consiste de um inteiro  $N$  que equivale à quantidade de clientes que chegaram à loja. As próximas  $N$  linhas contêm 3 informações cada: Um inteiro  $O$  representando o número da cliente, um inteiro  $P$  representando a quantidade de produtos que carrega e um inteiro  $Q$  representando a operação que deseja fazer (compra ou reparo).

## Saída:

A saída deve constar de apenas uma linha, com os números das clientes separados por espaços, da primeira que deve ser atendida até a última.

## Limites:

$$0 < N, O < 10^3$$

$$0 \leq P < 10^4$$

$$0 \leq Q \leq 1$$

## Exemplos:

| Entrada:            | Saída: |
|---------------------|--------|
| 1<br>1 1 0          | 1      |
| Entrada:            | Saída: |
| 2<br>1 2 1<br>2 4 0 | 1 2    |
| Entrada:            | Saída: |
| 3                   | 2 3 1  |

|                         |  |
|-------------------------|--|
| 1 3 1<br>2 5 0<br>3 3 0 |  |
|-------------------------|--|

*Autor: Lucas Valença*

## Problema J

Popojan é um aluno de Ciência da Computação que é apaixonado por esportes. Durante vários dias na semana, ele se reúne com amigos para bater uma “pelada”, seja numa quadra de futsal ou num society. Entretanto, essas peladas são durante a noite, fazendo com que ele chegue em casa bem cansado precisando acordar bem cedo no dia seguinte. Sendo assim, ele define uma hora pra acordar.

Popojan possui um relógio digital com despertador, que é o que lhe acorda. Um problema que o tem atormentado na hora de dormir é saber quantos minutos ele teria de sono se adormecesse imediatamente e acordasse somente quando o despertador tocasse. Mas ele realmente não é muito bom com números, e por isso pediu sua ajuda para escrever um programa que, dada a hora corrente e a hora do alarme, determine o número de minutos que ele poderá dormir.

### Entrada:

A entrada contém vários casos de teste. Cada caso de teste é descrito em uma linha, contendo quatro números inteiros H1, M1, H2 e M2, com H1:M1 representando a hora e minuto atuais, e H2:M2 representando a hora e minuto para os quais o alarme despertador foi programado ( $0 \leq H1 \leq 23$ ,  $0 \leq M1 \leq 59$ ,  $0 \leq H2 \leq 23$ ,  $0 \leq M2 \leq 59$ ).

O final da entrada é indicado por uma linha que contém apenas quatro zeros, separados por espaços em branco.

### Saída:

Para cada caso de teste da entrada seu programa deve imprimir uma linha, cada uma contendo um número inteiro, indicando o número de minutos que Popojan tem para dormir.

| Entrada:   | Saída:            |
|--|-------------------|
| 1 5 3 5<br>23 59 0 34<br>21 33 21 10<br>0 0 0 0      | 120<br>35<br>1417 |
| 22 38 01 37<br>00 00 05 45<br>22 00 07 31<br>0 0 0 0 | 179<br>345<br>571 |

Autor: Edjan Michiles