



IX - Maratona POP Médio – Le Dicário!

Olá pessoal, este é o Le Dicário! O objetivo deste documento é fazer com que a maratona POP seja, antes de um momento de competição, um momento de aprendizagem.

Deste modo, preparamos um conjunto de dicas sobre os problemas da prova, de modo a destravar algum competidor que esteja com problemas, ou mesmo criar a possibilidade de resolução de mais questões. Então vamos lá!

Ordem de dificuldade das questões:

As dicas que seguem são dadas em ordem crescente da dificuldade das questões. Isto é, segundo o nosso julgamento, as questões mais fáceis da prova aparecem aqui antes das mais difíceis.

Dicas gerais sobre Entrada e Saída

Use as funções padrão de leitura da sua linguagem favorita. Preparamos exemplos de leitura na questão X do aquecimento. Considere copiá-las.

Evite imprimir qualquer coisa que não seja pedido pelo problema. Por exemplo, nunca imprima algo como “digite um número inteiro:”, a não ser que o problema peça exatamente isso.

Dica sobre a questão E

Bom, se vocês estão enfrentando dificuldades com este problema, por favor, entrem em contato com a organização, eles irão até aí te ajudar! ==)

Dica sobre a questão A

Esta questão pede o domínio do laço do tipo **for**. Caso não tenha este domínio, que tal usar um laço do tipo **while**? Algo como, `x=20; WHILE x>=0 DO; PRINT x; DONE.`

Dica sobre a questão I

Acho q a maior dificuldade nesta questão é saber quando o presente cabe em uma caixa. Neste caso, basta comparar a maior, a do meio e a menor medida da caixa com a do presente. Caso a medida da caixa seja sempre maior ou igual que a do presente, ele cabe nesta caixa.

Dica sobre a questão C

Considere usar um vetor (*array* ou lista) para armazenar as pontuações, pois depois de calcular a média de todos os participantes, você pode usar o vetor para computar quantos alunos ficaram acima da média.

Lembre que a fórmula da média é a soma de todas as notas dividida pelo número de alunos. Cuidado, a média pode ser um número fracionário.

Dica sobre a questão B

Este problema trata do conhecimento sobre vetores multidimensionais (matrizes ou listas de listas) e sobre divisibilidade. Uma vez criada a matriz, deve-se varrer posição por posição para verificar qual é a mais vantajosa. Considere varrer a matriz variando primeiro a coluna e depois a linha, assim o desempate será o mesmo pedido pela questão. Lembre que um número x é divisível y se $x \% y == 0$;

Dica sobre a questão D

O problema pode ser resolvido usando a fórmula de permutação com repetição. Lembre-se que para calcular a quantidade de permutações de N elementos com A_1, A_2, \dots, A_k elementos repetidos, usamos $N! / (A_1! * A_2! * \dots * A_k!)$, onde a exclamação representa a função fatorial. No entanto, o problema quer que realizemos operações com módulo e aí as operações aritméticas mudam um pouco:

Adição: $(a + b) \% X$

Subtração: $(a - b) \% X$

Multiplicação: $(a * b) \% X$

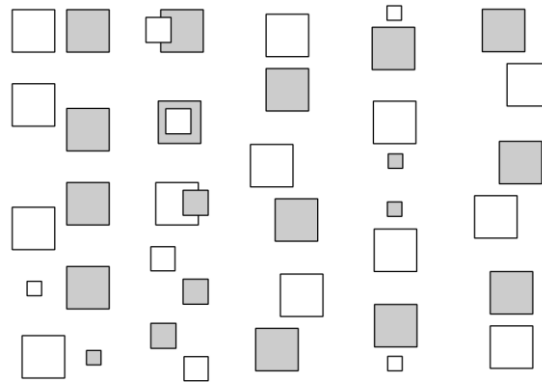
Divisão: $(a * \text{inv}(b)) \% X$

Então todas as operações devem ser feitas assim (inclusive na hora de calcular o fatorial). Ah, e note que o símbolo de porcentagem está representando o módulo ou resto da divisão.

Na divisão, também aparece uma função nova: **inv**. A função **inv** calcula o inverso modular de um valor e funciona da seguinte forma: **inv(k)** é um número entre **1** e **X - 1** que tem a propriedade de que $(k * \text{inv}(k)) \% X$ é igual a **1**. Como **X** é pequeno, é possível encontrar esse número com um laço.

Dica sobre a questão F

Não é tão difícil calcular a distância entre dois quadrados. Fica mais fácil sabendo que estes são paralelos, como na questão abordada. De um jeito simples, basta considerar todas as possibilidades (seguem algumas) e fazer vários IFs...



Dada a distância entre 2 quadrados, resta saber se há caminho a partir do quadrado de origem até o seu destino. Que tal criar um sistema de cores: preto, um quadrado que já foi visitado. Cinza um quadrado que já foi visto e Branco um quadrado que nunca foi visto nem visitado. No início, apenas o quadrado de origem é Cinza, todos os outros são brancos. Olha-se cada quadrado vizinho deste e os que forem brancos se tornam cinza. A partir de então, a origem fica preta, pois todos seus vizinhos ficaram cinza. Escolhe-se um quadrado cinza e repete o procedimento até que não restem quadrados cinza. Se o destino não ficou branco, há caminho entre eles. =-)

Dica sobre a questão H

Esta questão é um pouco mais complicada... Considere usar uma programação dinâmica com 3 dimensões, d, i, j , onde d seria o número de diagonais e i, j seriam o subproblema com uma grade menor (até a linha i e a coluna j). Ou seja: resolva o problema como se nenhuma diagonal pudesse ser usada. Use esta informação para resolvê-lo colocando no máximo 1 diagonal. Use esta informação para resolvê-lo colocando no máximo 2 diagonais. Assim até N diagonais, que é o máximo.

Dica sobre a questão G

A questão mais difícil da prova. Sinta-se desafiado a resolvê-la! Considere fazer uma BFS a partir de cada origem e salvar o resultado como a distância desta origem para todas as outras áreas. De posse deste resultado, basta combiná-los 3 a 3 para encontrar o menor número de dias... Bom, falando assim parece fácil. E é, mas depende de uma programação bem enxuta.

Boa sorte!