

Lista de Exercícios de Estruturas de Dados e Algoritmos

Q1) **[Problema 1514 do URI Online Judge]** A maioria dos programadores que chegam a escrever competições com exercícios de programação concordam em quatro características que toda competição deve alcançar. Embora nem todas sejam sempre alcançadas, quanto mais melhor. As características são as seguintes:

1. Ninguém resolveu todos os problemas.
2. Todo problema foi resolvido por pelo menos uma pessoa (não necessariamente a mesma).
3. Não há nenhum problema resolvido por todos.
4. Todos resolveram ao menos um problema (não necessariamente o mesmo).

Você organizou uma competição alguns dias atrás, e está preocupado com quantas dessas características ele conseguiu alcançar com a competição. Dadas as informações sobre a competição, com o número de participantes, número de problemas, e qual participante resolveu quais problemas, descubra o número de características que foram alcançadas nesta competição.

Haverá diversos casos de teste. Cada caso de teste inicia com dois inteiros **N** e **M** ($3 \leq N, M \leq 100$), indicando, respectivamente, o número de participantes e o número de problemas. Em seguida, haverá **N** linhas com **M** inteiros cada, onde o inteiro da linha **i** e coluna **j** é 1 caso o competidor **i** resolveu o problema **j**, ou 0 caso contrário. O último caso de teste é indicado quando **N = M = 0**, o qual não deverá ser processado.

Para cada caso de teste, imprima uma linha contendo um inteiro, representando quantas das características citadas foram alcançadas na competição.

Entrada	Saída
3 3 1 1 0 0 1 0 0 0 0 3 3 1 1 0 0 1 0 0 0 1 0 0	2 4

Q2) **[Problema 1171 do URI Online Judge]** Neste problema sua tarefa será ler vários números e em seguida dizer quantas vezes cada número aparece na entrada de dados, ou seja, deve-se escrever cada um dos valores distintos que aparecem na entrada por ordem crescente de valor.

A entrada contém apenas 1 caso de teste. A primeira linha de entrada contém um único inteiro **N**, que indica a quantidade de valores que serão lidos para **X** ($1 \leq X \leq 2000$) logo em seguida. Com certeza cada número não aparecerá mais do que 20 vezes na entrada de dados.

Imprima a saída de acordo com o exemplo fornecido abaixo, indicando quantas vezes cada um deles aparece na entrada por ordem crescente de valor.

Entrada	Saída
7 8 10 8 260 4 10 10	4 aparece 1 vez(es) 8 aparece 2 vez(es) 10 aparece 3 vez(es) 260 aparece 1 vez(es)

Q3) Sabendo-se as seguintes estruturas **typedef struct data{int mes, ano;} TD;** e

typedef struct prod{int cod; float peso, preco; TD *valid;} TPROD;, escreva a ordenação de **n** produtos na prateleira - **void ordena1 (TPROD **prat, int n);** - de acordo com os seguintes critérios:

1. em ordem crescente de código;
2. se dois produtos possuem o mesmo código, o critério de desempate é a data de validade, isto é, o produto que possui a data de validade mais próxima aparecerá primeiro na prateleira; e
3. se os empates persistirem, o critério de desempate é o preço, em ordem decrescente.

Q4) Refaça o exemplo feito em aula para usar **TCAND *vet[n]** e **TCAND **vet** no lugar de **TCAND vet[n]**.