

# [ED029] Confusão no Aeroporto

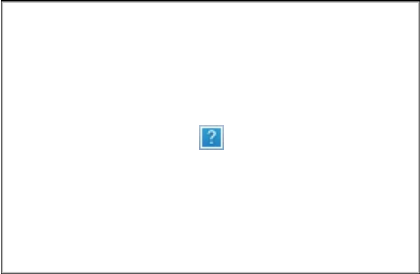
Neste problema deverá submeter uma classe **ED029** contendo um programa completo para resolver o problema (ou seja, com o método `main`).

Pode assumir que no Mooshak terá acesso às classes de listas, pilhas e filas como dadas nas aulas (ou seja, não precisa de as incluir no código submetido).

## O problema

Num país muito distante, o aeroporto de Earl Dude Anthony (E.D.A.) está a passar por uma fase de intenso tráfego. É que vai existir na cidade uma conferência sobre Estruturas de Dados e Algoritmos e ninguém quer perder esta oportunidade única.

O problema é que com tantos aviões alguns acabam por se atrasar, pois o aeroporto de E.D.A. só dispõe de uma única pista, que é utilizada por todos para aterrar ou levantar. O que o queríamos saber é mesmo qual o atraso que cada um dos aviões irá ter, sabendo inicialmente qual a hora prevista para cada um deles (todos os tempos serão dados em minutos).



Em cada momento, deve ser procurado o avião que está há mais tempo à espera (quer seja para descolar ou para aterrar), que deverá assim prosseguir a sua viagem. Em caso de empate, deve ser escolhido o avião que pretende aterrar (não queres ter problemas com aviões a ficar sem gasolina, pois não?). Deves também ter em conta que o processo de aterragem ou descolagem demora exactamente 1 minuto.

Por exemplo, imagina o seguinte cenário. Tens quatro aviões que querem descolar e três que querem aterrar:

Avião para descolar	Minuto previsto	Avião para aterrar	Minuto previsto
LEVANTA1	1	ATERRA1	1
LEVANTA2	2	ATERRA2	2
LEVANTA3	3	ATERRA3	3
LEVANTA4	4		

No minuto 1, dois aviões têm tarefas previstas (LEVANTA1 e ATERRA1). Como ambos estão ao mesmo tempo à espera (0 minutos) deve ser escolhido ATERRA1, que assim não sofre nenhum atraso. 1 minuto depois, com o avião já em terra, temos três aviões a quererem a pista (LEVANTA1, LEVANTA2 e ATERRA2). O que está há mais tempo à espera é o LEVANTA1 (1 minuto de espera) e por isso levanta vôo, com um atraso de 1 minuto. Este processo é continuado até todos os aviões terem conseguido a pista, registando-se os respectivos atrasos.

A tua tarefa é precisamente simular este processo e mostrar o atraso que cada avião teve no aeroporto de E.D.A..

## Input

Na primeira linha de input vem **C**, o número de casos a analisar.

De seguida, cada caso começa com uma linha com dois números inteiros, **L** e **A** (**L,A > 0**) representando respectivamente o número de aviões que querem levantar e o número de aviões que querem aterrar.

Seguem-se **L** linhas indicando os aviões que querem levantar, no formato "**NOME\_DE\_AVIÃO MINUTO\_DESEJADO**", onde o nome do avião é uma palavra constituída unicamente por letras e/ou números e o minuto desejado para levantar é um inteiro menor do que  $2^{31}$ . Depois vêm **A** linhas no mesmo formato, indicando os aviões que querem aterrar. Podes assumir que dentro de cada tipo de operação (aterragem e descolagem) os aviões são dados por ordem crescente do minuto desejado, sendo que esse mesmo minuto não aparece nunca repetido (nunca acontece, por exemplo, dois aviões quererem aterrar ao mesmo tempo).

Não existe nenhuma linha em branco a separar os vários casos de input.

## Output

Cada caso de output deve começar por ter uma linha com o número de aviões que querem levantar (**L**) e o número dos que querem aterrar (**A**), por esta ordem, e com um único espaço a separar os dois números.

De seguida devem vir **L** linhas, indicando o atraso dos aviões que querem levantar, no formato "**NOME\_DE\_AVIÃO**

**ATRASSO"**, sendo o atraso medido em minutos. Depois devem vir **A** linhas precisamente no mesmo formato, indicando os atrasos dos aviões que querem aterrar. Podes assumir que o atraso é um número menor do que 2^31.

Não devem existir linhas em branco a separar os diferentes casos.

**Exemplo de input/output**

Input	Output
2	1 1
1 1	TAP2345 1
TAP2345 1	IB2541 0
IB2541 1	4 3
4 3	LEVANTA1 1
LEVANTA1 1	LEVANTA2 2
LEVANTA2 2	LEVANTA3 3
LEVANTA3 3	LEVANTA4 3
LEVANTA4 4	ATERRA1 0
ATERRA1 1	ATERRA2 1
ATERRA2 2	ATERRA3 2
ATERRA3 3	