Pizza Instantânea

A pizzaria “É Pra Ontem” se orgulha de ter o sistema de entrega em domicílio mais rápido da cidade. Por outro lado, o sistema computacional deles não é tão bom assim. Trabalhar na “É Pra Ontem” não é uma tarefa fácil...

No sistema dessa pizzaria, cada cliente é identificado por uma ID numérica cujo valor é sempre positivo e nunca maior que 10000 (dez mil). Cada cliente sempre faz no máximo um pedido por dia e o mesmo cliente pode ser associado a mais de uma ID se fizer pedidos em dias distintos. A pizzaria nunca atende mais que 10000 clientes por dia. Um mesmo cliente pode solicitar várias pizzas no mesmo pedido.

Para cada pizza de sabor S solicitada pelo cliente que possui a ID de número C, o sistema adiciona uma entrada $X \rightarrow (C, S)$ em uma tabela hash. Quando o gerente quer listar todos os pedidos do dia, o sistema retorna uma lista de tuplas $T_i = (X_i, C_i, S_i)$ ordenadas pelas chaves X_i. Essa listagem é crucial para a operação da pizzaria.

A pizzaria é tão rápida que, num dia normal de trabalho, eles seguem um algoritmo de atendimento único:

1) Todos os clientes ligam e fazem seus pedidos;
2) Os pizzaiolos esquentam os fornos. Isso leva exatamente um minuto;
3) Todas as pizzas solicitadas pelo cliente com a menor ID do dia que ainda não foi atendido são produzidas e entregues num intervalo de tempo desprezível;
4) Todas as formas usadas para atender o cliente são lavadas. Um único estagiário leva um minuto para lavar cada forma. Os pizzaiolos descansam até a lavagem terminar;
5) O cliente atendido é marcado como atendido e, se ainda houver clientes esperando suas pizzas, volta-se ao passo 3.

Ao final do dia, os gerentes fazem perguntas do tipo “quanto tempo o cliente C esperou até receber suas pizzas?”. Inicialmente eles procuravam em toda a tabela hash pelas ID dos clientes. Depois de algum tempo, o sobrinho do dono da pizzaria escreveu um programa que fornecia apenas as ID, ordenadas pela chave referente na tabela hash, diminuindo um pouco o trabalho.

O problema é que, como o procedimento de cálculo da chave X é tão complexo, essa listagem de ID é feita em ordem pseudoaleatória. E como o número de pizzas produzidas em um único dia de atendimento é tão elevado, o sobrinho não conseguiu escrever um programa para ordenar essa listagem eficientemente.

É aí que você entra em cena. Você deverá escrever um programa que, dada a lista de ID em uma ordem qualquer, seja capaz de responder questionamentos do tipo “em quantos minutos o cliente C recebeu suas pizzas?”

Lembre-se: você deve ter sucesso exatamente onde o sobrinho falhou! Caso contrário você não receberá seu pagamento.

Entrada

A entrada é composta por vários casos de teste. Cada caso de teste começa com dois inteiros `N`, `Q` indicando o número de pizzas pedidas e o número de questionamentos que o sistema deverá responder.

Em seguida seguem-se `N` valores inteiros I_1, I_2, I_3, ..., I_N. Cada valor I_i é uma ID de um dos clientes que solicitaram pizzas naquele dia. Lembre-se que cada ID é um valor positivo e não-maior que 10000.

A isso seguem-se Q valores inteiros q_1, q_2, q_3, ..., q_Q. Para cada valor q_j você deve processar a consulta “quantos minutos o cliente q_j esperou até a chegada das pizzas?” Os valores q_j são inteiros positivos não-maiores que 10000.

A entrada termina quando N = 0 e Q = 0. Esse caso não deverá ser processado. Em todos os outros casos haverá ao menos um pedido e ao menos uma consulta.

O espaçamento entre casos de teste e valores poderá variar na entrada. Faça a leitura com `scanf()`. Não use `std::cin` para tentar resolver este problema ou você provavelmente receberá TIME LIMIT EXCEEDED.

Saída

Para cada caso de teste, imprima uma linha contendo o texto “Dia k:”. O primeiro caso de teste corresponde ao dia k = 1, o segundo caso corresponde ao dia k = 2 e assim por diante.

Em cada uma das j = 1, 2, ..., Q linhas seguintes, imprima o texto “Cliente q_j: $t_j$min.”, identificando que, naquele dia, o cliente com a ID q_j esperou t_j minutos para receber suas pizzas. Caso nenhum cliente com aquela ID tenha solicitado pizzas naquele dia, imprima uma linha contendo o texto: “Cliente q_j: quem?”.

Faça a escrita com `printf()`. Não use `std::cout` para tentar resolver este problema ou você provavelmente receberá TIME LIMIT EXCEEDED.

Exemplo de entrada e saída

Entrada	Saída esperada
4 3 3 2 1 1 1 2 3 4 1 3 2 1 1 6 6 2 10 8 6 2 6 6 5 6 0 0	Dia 1: Cliente 1: 1min. Cliente 2: 3min. Cliente 3: 4min. Dia 2: Cliente 6: quem? Dia 3: Cliente 5: quem? Cliente 6: 2min.