



Pontifícia Universidade Católica de Minas Gerais
Curso de Ciência da Computação
Disciplina: Algoritmos e Estruturas de Dados II

Laboratório 04 - Ordenação

Sort! Sort!! e Sort!!!

Hmm! Aqui você foi solicitado a fazer uma simples ordenação. A você serão dado N números e um inteiro positivo M . Você terá que ordenar estes N números em ordem ascendente de seu módulo M . Se houver um empate entre um número ímpar e um número par (para os quais o seu módulo M dá o mesmo valor) então o número ímpar irá preceder o número par. Se houver um empate entre dois números ímpares (para os quais o seu módulo M dá o mesmo valor), então o maior número ímpar irá preceder o menor número ímpar. Se houve um empate entre dois números pares (para os quais o seu módulo M dá o mesmo valor), então o menor número par irá preceder o maior número par. Para o resto de valores negativos siga a regra de linguagem de programação C: um número negativo nunca pode ter módulo maior do que zero. Por exemplo, $-100 \bmod 3 = -1$, $-100 \bmod 4 = 0$, etc.

Entrada

A entrada contém vários casos de teste. Cada caso de teste inicia com dois inteiros N ($0 < N < 10000$) e M ($0 < M < 10000$) que denotam quantos números existirão neste conjunto. Cada uma das próximas N linhas conterà um número cada. Estes números deverão caber em um inteiro de 32 bits com sinal. A entrada é terminada por uma linha que conterà dois valores nulos (0) e não deve ser processada.

Saída

A primeira linha de cada conjunto de saída irá contér os valores de N e M . As próximas N linhas irão contér N números, ordenados de acordo com as regras acima mencionadas. Imprima os dois últimos zeros da entrada para a saída padrão.

No seu código, realize uma análise da ordem de complexidade, detalhando o custo relacionado às comparações e movimentações conforme a estratégia utilizada. Toda a explicação deve ser apresentada no formato de comentário.

Exemplo de Entrada	Exemplo de Saída
15 3	15 3
1	15
2	9
3	3
4	6
5	12
6	13
7	7
8	1
9	4
10	10
11	11
12	5
13	2
14	8
15	14
3 3	3 3
9	9
12	12
10	10
0 0	0 0