Problema B
Rouba-Monte

Nome do arquivo fonte: `rouba.c`, `rouba.cpp`, `rouba.java` *ou* `rouba.pas`

Um dos jogos de cartas mais divertidos para crianças pequenas, pela simplicidade, é Rouba-Monte. O jogo utiliza um ou mais baralhos normais e tem regras muito simples. Cartas são distingüidas apenas pelo valor (ás, dois, três, ...), ou seja, os naipes das cartas não são considerados (por exemplo, ás de paus e ás de ouro têm o mesmo valor).

Inicialmente as cartas são embaralhadas e colocadas em uma pilha na mesa de jogo, chamada de pilha de compra, com face voltada para baixo. Durante o jogo, cada jogador mantém um *monte* de cartas, com face voltada para cima; em um dado momento o monte de um jogador pode conter zero ou mais cartas. No início do jogo, todos os montes dos jogadores têm zero cartas. Ao lado da pilha de compras encontra-se uma área denominada de *área de descarte*, inicialmente vazia, e todas as cartas colocadas na área de descarte são colocadas lado a lado com a face para cima (ou seja, não são empilhadas).

Os jogadores, dispostos em um círculo ao redor da mesa de jogo, jogam em sequência, em sentido horário. As jogadas prosseguem da seguinte forma:

- O jogador que tem a vez de jogar retira a carta de cima da pilha de compras e a mostra aos outros jogadores; vamos chamar essa carta de *carta da vez*.
- Se a carta da vez for igual a alguma carta presente na área de descarte, o jogador retira essa carta da área de descarte colocando-a, juntamente com a carta da vez, no topo de seu monte, com as faces voltada para cima, e continua a jogada (ou seja, retira outra carta da pilha de compras e repete o processo).
- Se a carta da vez for igual à carta de cima de um monte de um outro jogador, o jogador "rouba" esse monte, empilhando-o em seu próprio monte, coloca a carta da vez no topo do seu monte, face para cima, e continua a jogada.
- Se a carta da vez for igual à carta no topo de seu próprio monte, o jogador coloca a carta da vez no topo de seu próprio monte, com a face para cima, e continua a jogada.
- Se a carta da vez for diferente das cartas da área de descarte e das cartas nos topos dos montes, o jogador a coloca na área de descarte, face para cima, e a jogada se encerra (ou seja, o próximo jogador efetua a sua jogada). Note que esse é o único caso em que o jogador não continua a jogada.

O jogo termina quando não há mais cartas na pilha de compras. O jogador que tiver o maior monte (o monte contendo o maior número de cartas) ganha o jogo. Se houver empate, todos os jogadores com o monte contendo o maior número de cartas ganham o jogo.

Entrada

A entrada é composta de vários casos de teste. A primeira linha de um caso de teste contém dois inteiros N e J, representando respectivamente o número de cartas no baralho ($2 \leq N \leq 10.000$)

e o número de jogadores ($2 \leq J \leq 20$ e $J \leq N$). As cartas do baralho são representadas por números inteiros de 1 a 13 e os jogadores são identificados por inteiros de 1 a J. O primeiro jogador a jogar é o de número 1, seguido no jogador de número 2, ..., seguido pelo jogador de número J, seguido pelo jogador de número 1, seguido do jogador de número 2, e assim por diante enquanto houver cartas na pilha de compras. A segunda linha de um caso de teste contém N inteiros entre 1 e 13, separados por um espaço em branco, representando as cartas na pilha de compras. As cartas são retiradas da pilha de compras na ordem em que aparecem na entrada. O final da entrada é indicado por uma linha com $N = J = 0$.

A entrada deve ser lida da entrada padrão.

Saída

Para cada caso de teste seu programa deve imprimir uma linha, contendo o número de cartas do monte do jogador ou jogadores que ganharam a partida, seguido de um espaço em branco, seguido do(s) identificador(es) dos jogadores que ganharam a partida. Se há mais de um jogador vencedor imprima os identificadores dos jogadores em ordem crescente, separados por um espaço em branco.

A saída deve ser escrita na saída padrão.

Exemplo de entrada	**Saída para o exemplo de entrada**

`4 2`
`10 7 2 5`
`6 3`
`1 2 1 2 1 2`
`8 2`
`3 3 1 1 2 3 4 5`
`0 0`
	`0 1 2`
	`5 1`
	`3 2`