

# Vivo ou Morto

Arquivo fonte: *vivo.c*, *vivo.cc*, *vivo.cpp* ou *vivo.pas*

Toda criança certamente já brincou de “vivo ou morto”. A brincadeira é dirigida por um “chefe” (um adulto), que comanda dois ou mais participantes (crianças). A brincadeira é composta de rodadas. No início, os participantes são organizados pelo chefe em fila única. A cada rodada o chefe grita “vivo” ou “morto” e todos os participantes tentam seguir sua ordem, levantando-se ao ouvir a palavra “vivo” ou abaixando-se ao ouvir a palavra “morto”. Um participante que não segue a ordem do chefe é eliminado, deixando o seu lugar na fila. Os participantes remanescentes agrupam-se novamente em fila única, preenchendo as posições dos participantes eliminados, mas mantendo suas posições relativas. O jogo continua até que uma rodada seja composta por exatamente um participante. Tal participante é dito o vencedor do jogo.

Por exemplo, considere que a brincadeira inicie com cinco participantes, identificados por números inteiros de 1 a 5, e que o chefe organize a fila na ordem  $3 \rightarrow 2 \rightarrow 1 \rightarrow 4 \rightarrow 5$ . Se na primeira rodada forem eliminados os participantes 2 e 4, a fila da segunda rodada será formada por  $3 \rightarrow 1 \rightarrow 5$ ; se na segunda rodada for eliminado o participante 1, a fila da terceira rodada será formada por  $3 \rightarrow 5$ . Se na terceira rodada o participante 3 for eliminado, o vencedor da brincadeira será o participante 5.

## Tarefa

Sua tarefa é escrever um programa que determine o vencedor de uma partida de “vivo ou morto”, a partir da informação das ordens dadas pelo chefe e das ações executadas pelos participantes em cada rodada.

## Entrada

A entrada é constituída de vários casos de teste, cada um representando uma partida. A primeira linha de um caso de teste contém dois números inteiros  $P$  e  $R$  indicando respectivamente a quantidade inicial de participantes ( $2 \leq P \leq 100$ ) e quantidade de rodadas da partida ( $1 \leq R \leq 100$ ). Os participantes são identificados por números de 1 a  $P$ . A segunda linha de um caso de teste descreve a fila organizada pelo chefe, contendo  $P$  números inteiros distintos  $x_1, x_2, \dots, x_P$ , onde  $x_1$  representa o identificador do participante no primeiro lugar na fila,  $x_2$  representa o identificador do participante no segundo lugar na fila, e assim por diante ( $1 \leq x_i \leq P$ ). Cada uma das  $R$  linhas seguintes representa uma rodada, contendo um número inteiro  $N$  indicando o número de participantes da rodada ( $2 \leq N \leq P$ ), um número inteiro  $J$  representando a ordem dada pelo chefe ( $0 \leq J \leq 1$ ) e  $N$  números inteiros  $A_i$  representando a ação do participante colocado na  $i$ -ésima posição na fila ( $0 \leq A_i \leq 1$ ). Ordens e ações “vivo” são representadas pelo valor 1, ordens e ações “morto” pelo valor zero. Cada partida tem exatamente um vencedor, determinado somente na última rodada fornecida no caso de teste correspondente. O final da entrada é indicado por  $P = R = 0$ .

*A entrada deve ser lida do dispositivo de entrada padrão (normalmente o teclado).*

## Saída

Para cada caso de teste seu programa deve produzir três linhas. A primeira identifica o conjunto de teste no formato “**Teste n**”, onde **n** é numerado a partir de 1. A segunda linha deve conter o identificador do vencedor. A terceira linha deve ser deixada em branco. A grafia mostrada no Exemplo de Saída, abaixo, deve ser seguida rigorosamente.

*A saída deve ser escrita no dispositivo de saída padrão (normalmente a tela).*

## Restrições

$2 \leq P \leq 100$  ( $P = 0$  apenas para indicar o fim da entrada)

$1 \leq R \leq 100$  ( $R = 0$  apenas para indicar o fim da entrada)

$1 \leq x_i \leq P$ , para  $1 \leq i \leq P$

$2 \leq N \leq P$

$0 \leq J \leq 1$

$0 \leq A_i \leq 1$ , para  $1 \leq i \leq N$

Exemplo de Entrada	Saída para o Exemplo de Entrada
2 2	Teste 1
2 1	2
2 1 1 1	
2 1 1 0	Teste 2
5 4	5
3 2 1 4 5	
5 1 1 1 1 1 1	
5 0 0 1 0 1 0	
3 0 0 1 0	
2 1 0 1	
0 0	