

Problema A: Prague vikings?

Arquivo: vikings.[c|cpp|pas|java]

Vestígios de uma antiga civilização viking foram descobertos nos arredores de Praga, e uma grande quantidade de material impresso foi achada junto ao sítio arqueológico. Como esperado, a leitura deste material mostrou-se uma tarefa árdua e desafiadora, já que essa civilização utilizava um esquema de codificação de texto para evitar que seu conhecimento fosse assimilado por seus rivais.

Recentemente, pesquisadores tchecos anunciaram com grande euforia à imprensa a compreensão do mecanismo de codificação utilizado por esses vikings. De acordo com os pesquisadores, o alfabeto viking era composto pelas letras de *A* até *Z* (incluindo as letras *K*, *W* e *Y*).

A codificação era realizada da forma que segue. Inicialmente, era construída uma lista em que a letra *A* aparecia na primeira posição, a letra *B* aparecia na segunda, e assim sucessivamente, com as letras seguindo a mesma ordem que em nosso alfabeto. Em seguida, o texto a ser codificado era varrido da esquerda para a direita e, para cada letra *l* encontrada, o número de sua posição na lista era impresso e *l* era movida para o início da lista. Por exemplo, a codificação viking para a mensagem:

A B B B A A B B B B A C C A B B A A A B C

era dada pela seguinte seqüência de inteiros:

1 2 1 1 2 1 2 1 1 1 2 3 1 2 3 1 2 1 1 2 3

Os pesquisadores tchecos pediram sua ajuda para construir um programa que receba uma seqüência de inteiros que representa uma mensagem codificada e decodifica-a.

Entrada

Seu programa deve estar preparado para trabalhar com diversas instâncias. Cada instância tem a estrutura que segue. Na primeira linha é fornecido um inteiro *m* ($0 \leq m \leq 10000$) que representa o número de inteiros que compõem o texto codificado. Na próxima linha são dados, separados por espaços em branco, os *m* valores inteiros (cada valor é maior ou igual a 1 e menor ou igual a 26). Um valor *m* = 0 indica o final das instâncias e não deve ser processado.

Saída

Para cada instância solucionada, você deverá imprimir um identificador **Instancia h** em que **h** é um número inteiro, seqüencial e crescente a partir de 1. Na linha seguinte, você deve imprimir o texto decodificado. Uma linha em branco deve separar a saída de cada instância.

Exemplo

Entrada

```
21
1 2 1 1 2 1 2 1 1 1 2 3 1 2 3 1 2 1 1 2 3
5
22 6 8 4 15
3
24 1 1
26
22 10 6 4 13 16 16 12 5 1 4 20 1 21 21 5 10 7 16 6 15 12 5 3 8 9
0
```

Saída

Instancia 1

ABBBAABBBBACCABBAABC

Instancia 2

VEGAN

Instancia 3

XXX

Instancia 4

VIDALONGAAOSSSTRAIGHTEDGERS