

Problema: Stream com contagem de quadros

Elaborado por: Rafael Lima

Um *stream* é um fluxo contínuo e sequencial de dados. Para usar um *stream* é necessário ter um sistema de *buffer* que consiste em uma estrutura de dados que recebe os dados do *stream* e ao receber, no momento apropriado, exibe-os. Por exemplo, no *stream* de vídeo, há um *buffer* recebendo o vídeo e passando para o reprodutor de conteúdo para ser exibido.

Sua missão é implementar um *buffer* de um *stream* de texto usando lista estática. Seu *buffer* terá a capacidade máxima de N caracteres, onde o número N será informado na primeira linha da entrada. A cada vez que a lista se tornar cheia é necessário imprimir os caracteres armazenados e esvaziar o *buffer* para recepcionar os demais caracteres. Cada caractere de entrada é seguido do respectivo número do quadro. Este número é um inteiro positivo e único para cada caractere. Devido a oscilações de conexão, os quadros não chegam em ordem, portanto os caracteres podem estar fora de ordem. Para isso, em cada evento de exibir (esvaziar o *buffer*) precisa exibir os caracteres ordenados pelo número do quadro.

A depender do tamanho do *buffer* e da desorganização no envio dos dados, pode ser que o *buffer* de N caracteres seja insuficiente para lidar com isso. Neste caso, pelo menos um pedaço da frase estará com caracteres fora do lugar. Portanto, é preciso detectar se houve perda de ordem por causa do tamanho do *buffer*. Essa informação deverá ser informada após a impressão de toda a frase.

Portanto, sua tarefa é fazer um programa em linguagem C, usando uma lista estática, capaz de armazenar no máximo N caracteres e implementar a funcionalidade de *buffer* com detecção de perda de sequenciamento de quadros.

Entrada

A primeira linha contém um inteiro N informando o tamanho máximo da lista estática. Cada linha a seguir é composta de um caractere e um número inteiro positivo e único identificando o número do quadro. A última leitura é o par 0 0, que não faz parte dos dados, apenas indica o final do *stream*.

Saída

Deve ser a frase da melhor maneira ordenada possível, considerando a ordenação pelo números do quadros e a limitação do tamanho do *buffer*; e em outra linha apresentar a mensagem sincronizado ou não sincronizado, conforme a frase tenha sido recomposta corretamente. Perceba que o **Exemplo de entrada 1** a seguir, com o tamanho do *buffer* é 10, não foi possível ordenar a frase para ficar sincronizada. Porém, ao aumentar o tamanho do *buffer* em uma unidade (Exemplo de entrada 2), já foi possível exibir a frase sem perda de quadros.

Exemplos

<p>Exemplo de entrada 1</p> <pre> 10 U 1 a 3 m 2 4 10 r 6 q 11 s 8 e 9 f 5 a 7 u 12 a 13 l 14 q 15 u 16 e 17 r 18 ! 19 0 0 </pre>	<p>Exemplo de saída 1</p> <pre> Uma frase qualquer! nao sincronizado </pre>
---	---

<p>Exemplo de entrada 2</p> <pre> 11 U 1 a 3 m 2 4 10 r 6 q 11 s 8 e 9 f 5 a 7 u 12 a 13 l 14 q 15 u 16 e 17 r 18 ! 19 0 0 </pre>	<p>Exemplo de saída 2</p> <pre> Uma frase qualquer! sincronizado </pre>
---	---

Exemplo de entrada 2	Exemplo de saída 2
30 a 3 e 2 C 8 j 4 a 5 ! 6 a 9 R 1 7 d 10 m 14 12 u 13 m 18 15 b 16 e 17 a 11 - 19 t 24 v 21 e 22 n 23 a 20 a 27 r 26 u 25 d 28 o 29 c 31 m 33 36 30 a 35 34 o 32 v 37 i 38 s 39 t 40 a 44 42 o 47 a 41 45 m 46 d 43 n 48 t 49 a 50 u 56 h 52 54 n 51 q 55 a 53	Reaja! Cada um bem-aventurado com a vista da montanha que sou muito diligente. nao sincronizado

e 57	
i 66	
s 59	
u 60	
58	
i 62	
r 63	
64	
d 65	
b 61	
l 67	
t 72	
g 69	
e 70	
n 71	
i 68	
e 73	
m 74	
e 75	
n 76	
t 77	
e 78	
. 79	
0 0	