

# intercala8

Um animado aluno está criando um sistema centralizado que recebe submissões de problemas, no estilo do MOJ, porém ele recebe submissões de diversas regiões do país.

O interessante é que o sistema processa as submissões a cada 5 minutos e por isso recebe muitos dados. Cada região envia um vetor ordenado pelo tempo decorrido da prova em que a submissão foi feita, ou seja, se uma submissão foi feita após 5 minutos de prova ela terá valor 5, se for 15 minutos, será 15.

O sistema deverá processar todas submissões, de todas as regiões. A ordem de processamento das submissões deverá ser ordenada pelo tempo decorrido da prova, ou seja, a submissão mais antiga (menor valor de tempo decorrido da prova), independente da região, é a que deverá ser feita primeiro.

O problema é relativamente simples, pois como todos os vetores recebidos já estão ordenados, basta criar um algoritmo que junte (intercale) todos os vetores em um único vetor.

## Entrada

A entrada é composta por único caso de teste, contendo até 16 linhas. A primeira linha, do caso de teste, possui um número inteiro  $S_i$  ( $0 \leq S_i \leq 100000$ ), na segunda linha haverão  $S_i$  inteiros  $N_s$  ( $0 \leq 1000000$ ), este padrão de 2 linhas é repetido 8 vezes. Veja os exemplos para melhor entender o conjunto de entrada e saída.

## Saída

A saída é composta por uma única linha contendo o resultado da intercalação dos 8 vetores.

*Dica:* Você pode achar conveniente implementar a função `intercala2`. É possível criar *wrappers*?

## Exemplo

### Exemplo de entrada

```
7
150642 277374 434157 450646 473069 503877 542425
0
7
13699 71698 231910 285705 320146 444689 527669
5
109819 277731 286399 614494 839443
8
27803 37390 124478 335586 515253 576101 604613 614748
3
127682 270851 304169
4
119953 303827 358184 826438
8
115223 383541 511235 631310 679285 707705 929900 983246
```

### Saída para o exemplo acima

```
13699 27803 37390 71698 109819 115223 119953 124478 127682 150642 231910 270851 277374 277731 285705 286399
```

### Exemplo de entrada

```
2
637492 795417
2
587713 734272
```

9  
192782 568692 586540 648744 688445 843307 861971 932683 985397  
1  
494715  
0  
5  
39172 84317 298036 723373 924454  
3  
321640 702103 897112  
3  
140100 196776 693775

**Saída para o exemplo acima**

39172 84317 140100 192782 196776 298036 321640 494715 568692 586540 587713 637492 648744 688445 693775 7021

*Author: Bruno Ribas*