

- **Prazo para submissão:** Final do semestre
- **Peso da atividade:** 1

Caos nas Filas do Banco

Atividade

Era um dia ensolarado de verão e o banco estava movimentado como sempre. As pessoas esperavam pacientemente na fila para serem atendidas pelos quatro caixas disponíveis. De repente, três caixas decidiram almoçar simultaneamente, e pararam de atender, deixando apenas um único funcionário para lidar com a multidão de clientes.



O gerente, desesperado, correu pelos corredores procurando ajuda. Foi então que ele lhe encontrou, um de seus melhores funcionários, conhecido por sua capacidade computacional para solução de problemas. O gerente pediu a sua ajuda para reorganizar as filas de modo a garantir que os clientes não ficassem irritados e o atendimento fosse otimizado.

Você rapidamente percebeu que a solução ideal seria criar um algoritmo de intercalação das quatro filas anteriores em uma só. Com sorte, os clientes possuíam uma ficha que indicava o tempo previsto de espera para a sua fila original, permitindo essa intercalação de maneira fácil. Além disso, percebendo que essa situação poderia se repetir no futuro, você decidiu criar uma solução que abstrai a quantidade de filas intercaladas para se transformar em uma só.

Computacionalmente, as filas são representadas por um conjunto de vetores inteiros de tamanho variável. Perceba que isso é similar a uma matriz, com a diferença que a quantidade de colunas não é uniforme para cada linha. Cada linha dessa matriz representa uma fila de um caixa diferente, e os elementos representam o tempo previsto de espera de cada cliente na sua fila original. A sua tarefa consiste em intercalar, linha a linha, os elementos desse conjunto de vetores em um único vetor. Esse vetor resultante contém a ordem de atendimento dos clientes na fila do único caixa atendente.

Entrada

A entrada é composta de:

- um número inteiro n na primeira linha, que indica a quantidade de filas originais;
- seguida de n linhas, cada uma com valores inteiros. O primeiro valor k de cada uma dessas linhas indica a quantidade de clientes naquela fila. Os próximos k valores **estão sempre ordenados** e indicam o tempo de espera de cada cliente naquela fila específica;

```
1 4
2 3 1 5 8
3 2 2 6
4 4 2 4 7 9
5 1 10
```

Nesse exemplo, o valor de n é igual a 4, já que existem quatro filas originais. O valor de k para a primeira fila é 3, já que existem 3 clientes nessa fila. O primeiro cliente possui um tempo de espera previsto de um minuto, o segundo de 5 minutos e o terceiro de 8 minutos.

Assuma que:

- A quantidade máxima de filas originais (n) é igual a 100;
- A quantidade máxima de clientes por fila (k é igual a 100;
- Pode haver clientes em filas diferentes com o mesmo tempo de espera. Nesse caso, não importa qual dos dois deve ser atendido primeiro na nova fila única.

Casos especiais

- Assuma que pode haver filas que possuem caixa atendendo, mas que nenhum cliente se encontra nela. Nesse caso, o valor de k será igual a 0.
- Assuma que pode haver uma quantidade de filas originais (n) igual a 0.

Saída

A saída é uma única linha representando a nova fila com os clientes intercalados de maneira justa. No caso do exemplo, a saída é:

```
1 1 2 2 4 5 6 7 8 9 10
```

mantendo a nova fila ordenada pelos tempos de espera originais.

Orientações Importantes

- É preciso tomar cuidado em relação à formatação da saída do programa, de modo a garantir o funcionamento com os casos de testes. Nessa atividade, observe que não há o caractere de nova linha "`\n`" ao final da saída. Seu programa deve imprimir apenas uma única linha com a nova fila ordenada.
- Você pode utilizar os casos de testes disponibilizados para entender a formatação correta da saída.
- Atente para o nome do arquivo entregue na plataforma do *runcodes*. A própria plataforma irá indicar o nome dos arquivos obrigatórios na hora da entrega. Nesse laboratório 1, um único arquivo de nome `FilaBancos.c` deve ser entregue.

Dicas

- Utilize as funções apresentadas de alocação dinâmica (***CriaVetor***, ***DestroiVetor***, ...) para gerenciar corretamente o tamanho de cada fila na memória.
- Um conjunto de vetores inteiros pode ser implementado por um **ponteiro duplo**. É possível alocar inicialmente a quantidade de vetores para os quais esse ponteiro aponta (quantidade de filas originais); e, em seguida, alocar a quantidade de elementos de cada fila.
- Você pode utilizar o elemento inicial desses ponteiros para guardar tanto a quantidade de filas originais (***n***) quanto a quantidade de clientes em cada fila (***k***), facilitando a implementação da solução.