

MC202 — ESTRUTURAS DE DADOS

Laboratório 10 — Intercalar k vetores ordenados

Tarefa

São dados k vetores de inteiros já ordenados, sendo que o número total de elementos em todos os vetores é m . Escreva uma função para devolver um único novo vetor ordenado com todos os m elementos. O tempo de execução do algoritmo deve ser $O(m \log k)$.

Entrada

Entrada é composta por um número k ($1 \leq k \leq 10^2$) que representa o número de vetores, seguido por k linhas, onde a i -ésima linha contém um inteiro v_i ($1 \leq v_i \leq 10^4$) que representa o tamanho do vetor i . Depois, haverá mais k linhas, onde a j -ésima linha contém v_j inteiros que são os valores do vetor j .

Saída

Uma linha contendo os valores de um único vetor ordenado composto por todos os valores dos k vetores de entrada, os valores são separados por espaço e há um caractere de final de linha após o último valor.

Exemplo

Entrada

```
3
3
2
4
1 3 6
4 5
8 10 12 13
```

Saída

```
1 3 4 5 6 8 10 12 13
```

Entrada

```
2
1
1
1
1
1
```

Saída

```
1 1
```

Entrada

```
2
5
5
1 3 5 7 9
0 2 4 6 8
```

Saída

```
0 1 2 3 4 5 6 7 8 9
```

Critérios específicos

- Você deverá implementar e utilizar uma fila de prioridade para esse lab
- Deverão ser submetidos os seguintes arquivos:
 - fila_de_prioridade.c (implementação da fila de prioridade)
 - fila_de_prioridade.h (interface da fila de prioridade)
 - lab10.c
- Tempo máximo de execução: 3 segundos.

Testando

Para compilar usando o Makefile fornecido e verificar se a solução está correta basta seguir o exemplo abaixo.

```
make
./lab10 < arq01.in > arq01.out
diff arq01.out arq01.res
```

onde arq01.in é a entrada (casos de testes disponíveis no SuSy) e arq01.out é a saída do seu programa. O Makefile também contém uma regra para testar todos os testes de uma vez; nesse caso, basta digitar:

```
make testar_tudo
```

Isso testará o seu programa com os casos abertos. Após o prazo, os casos de teste fechados serão liberados e podem ser baixados digitando-se:

```
make baixar_fechados
```

Para ver quanto tempo seu programa demora em cada teste, digite:

```
make tempo
```

Observações gerais

No SuSy, haverá 3 tipos de tarefas com siglas diferentes para cada laboratório de programação. Todas possuirão os mesmos casos de teste. As siglas são:

1. **DRAFT:** Esta tarefa serve para testar o programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém arquivos submetidos aqui não serão corrigidos.
2. **ENTREGA:** Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final dentro do prazo estabelecido para o laboratório. Não use essa tarefa para testar o seu programa: submeta aqui quando não for mais fazer alterações no seu programa.
3. **FORAPRAZO:** Esta tarefa tem limite de *uma única submissão* e serve para entregar a versão final após o prazo estabelecido para o laboratório, mas com nota reduzida (conforme a ementa). O envio nesta tarefa irá substituir a nota obtida na tarefa ENTREGA apenas se o aluno tiver realizado as correções sugeridas no feedback ou caso não tenha enviado anteriormente em ENTREGA.

Observações sobre SuSy:

- Versão do GCC: gcc (GCC) 4.8.5 20150623 (Red Hat 4.8.5-36)
- Flags de compilação:
-std=c99 -Wall -Werror -Werror=vla -pedantic-errors -g -lm

Além das observações acima, esse laboratório será avaliado pelos critérios gerais:

- Indentação de código e outras boas práticas, tais como:

- uso de comentários (úteis e apenas quando forem relevantes);
 - código simples e fácil de entender;
 - sem duplicidade (partes que fazem a mesma coisa).
- Organização do código:
 - tipos de dados criados pelo usuário e funções bem definidas e tão independentes quanto possível.
- Corretude do programa:
 - programa correto e implementado conforme solicitado no enunciado;
 - inicialização de variáveis sempre que for necessário;
 - dentre outros critérios.