

# MC202E - ESTRUTURAS DE DADOS

## Lab09: Inspeção

Um funcionário está encarregado da inspeção dos lotes de produtos que chegam na empresa. Como a quantidade é muito grande e os recursos são limitados, deve-se escolher um grupo de caixas de cada lote para serem inspecionadas. Se alguma das caixas inspecionadas for rejeitada, então todo o lote é rejeitado. Para escolher as caixas que serão inspecionadas, é necessário criar um sistema que faça a escolha usando um critério específico. A empresa decidiu testar muitos critérios diferentes e você recebeu a tarefa de escolher as caixas usando a mediana estatística.

Nesse método, caixas são adicionadas a um grupo para inspeção, onde uma ou duas caixas do grupo são selecionada para inspeção. As caixas selecionadas serão aquela cujo seu código estiver no meio de todos os códigos do grupo, i.e., será a mediana das caixas quando elas são ordenadas por código de identificação.

## Tarefa

Escreva um programa em C que recebe como entrada vários grupos de caixas para inspeção e que imprima, para cada grupo, o código de identificação das caixas selecionadas. O seu programa deverá utilizar a estrutura de dados **heap** para determinar quais caixas serão inspecionadas.

## Entrada

A primeira linha da entrada contém um inteiro positivo ***n*** que corresponde ao número de grupos para inspeção. Para cada grupo, a primeira linha contém um inteiro ***m*** que corresponde ao número de caixas do grupo. Cada uma das ***m*** linhas seguintes contém apenas o código de identificação da caixa. O código de identificação é uma *string* de 4 caracteres alfanuméricos (sem espaços em branco).

### Restrição

- $5 \leq m$

### Observação

- Considere que os códigos de identificação devem ser ordenados segundo a **ordem lexicográfica** do padrão ASCII.

# Saída

Para cada grupo a ser inspecionado, seu programa deverá imprimir uma linha contendo o identificador do grupo, seguindo da caixa ou das caixas selecionadas para inspeção segundo a mediana estatística. Considere que cada grupo é identificado em ordem crescente, a partir de 1.

## Observação

- ❑ Se grupo conter número ímpar de caixas, então há apenas uma caixa no meio de todos os códigos do grupo; caso contrário, há exatamente duas caixas.

# Exemplo

A grafia da saída abaixo deve ser seguida rigorosamente por seu programa, inclusive a impressão de uma linha em branco no final da saída.

## Entrada

```
2
5
K50N
PPME
M5FE
WYVD
EZDD
8
DUN5
S0V0
6F5W
42I7
P11Y
AYHK
RKD8
172Q
```

## Saída

```
1 M5FE
2 AYHK DUN5
```

## Dica

- ❑ Usar um objeto de **heap máxima** e **heap mínima**.

## Critérios específicos

Os seguintes critérios específicos sobre o envio, implementação, compilação e execução devem ser satisfeitos.

i. Submeter no SuSy os arquivos:

### Obrigatórios

⇒ `lab09.c`: Deverá conter a função principal para a solução do problema.

⇒ `heap.*`: Arquivos de cabeçalho e fonte devem conter (somente) a interface e implementação de uma estrutura de heap, cuja inserção e remoção tenha complexidade da ordem  $O(\lg m)$ . **Duplicidade de código não será permitido**, ou seja, implementar **apenas uma** função de inserção, remoção, etc.

### Opcionais

⇒ `*.*`: Enviar até 2 arquivos cabeçalho e 2 arquivos fonte, desde de que contribuam para a modularização da solução.

ii. É **obrigatório** implementar uma solução que utiliza **heap**. A heap **deverá ser genérica** em termos da comparação de elementos. Para isso, poderá manter em sua estrutura um ponteiro para uma função de comparação.

iii. Flags de compilação:

`-std=c99 -Wall -Werror -g -lm`

iv. Tempo máximo de execução: **1 segundo**.

## Observações gerais

No decorrer do semestre haverá 3 tipos de tarefas no SuSy (descritas logo abaixo). As tarefas possuirão os mesmos casos de testes abertos e fechados, no entanto o número de submissões permitidas e prazos são diferentes. As seguintes tarefas estão disponíveis no SuSy:

- ❑ **Lab09-AmbienteDeTeste**: Esta tarefa serve para testar seu programa no SuSy antes de submeter a versão final. Nessa tarefa, tanto o prazo quanto o número de submissões são ilimitados, porém os arquivos submetidos aqui **não serão corrigidos**.
- ❑ **Lab09-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 e submeta aqui apenas quando não for mais fazer alterações no seu programa.
- ❑ **Lab09-ForaDoPrazo**: Esta tarefa tem limite de uma **única** submissão e serve para entregar a versão final fora prazo estabelecido para o laboratório. Esta tarefa irá substituir a nota obtida na tarefa **Lab09-Entrega** apenas se o aluno tiver

realizado as correções sugeridas no *feedback* ou caso não tenha enviado anteriormente na tarefa **Lab09-Entrega**.

### **Avaliação**

Este laboratório será avaliado conforme o número de **casos de teste fechados** em que o seu programa apresentou saída correta, menos possíveis descontos referentes aos critérios de correção e de qualidade de código, os quais estão disponíveis na [planilha de notas](#). Entretanto, outros critérios podem ser incorporados na avaliação desta tarefa se for julgado pertinente; e **a nota pode ser zerada caso não atender os critérios específicos**.