

## 6º Exercício: Agenda Eletrônica

Elaborar um programa que implementa **fila de prioridade** para ajudar na organização de tarefas de uma agenda eletrônica.

O usuário insere as suas atividades na agenda e atribui a cada uma delas um valor de importância, no mínimo 1 e no máximo 99. As tarefas podem ser inseridas a qualquer momento do dia. Quando o usuário termina uma tarefa, clica na agenda para consultar qual a tarefa mais importante a ser realizada em seguida.

Ao terminar o expediente, a agenda permite que o usuário selecione o comando “fim”. Nessa ocasião, o programa deve desfazer a fila e imprimir, para consulta, todas as tarefas remanescentes, ordenadas por importância.

A programação deve ser da seguinte forma:

- Usar heap, implementado com vetor.
- A variável do heap deve ser uma `struct` contendo:
  - o vetor;
  - a quantidade de elementos;
  - a quantidade máxima possível de elementos.
- Fazer funções separadas para cada operação no heap:  
Insert, Extract, Inicializar, Finalizar, HeapVazio, HeapCheio.
- O heap deve ser acessado **apenas** através das funções acima!!
- As funções HeapVazio e HeapCheio retornam boolean.
- A função Finalizar deve somente dar `free`.
- Não usar variáveis globais.
- Apenas a função `main` pode ler e/ou escrever nos arquivos de entrada e saída.

Apresente na saída:

Mensagens de aviso:

- se o usuário consultar a próxima tarefa e o heap estiver vazio;
- se no final a agenda estiver vazia;

Mensagem de erro:

- se houver tentativa de inserir mais elementos que a capacidade do heap.

As mensagens são livres. Mas devem estar alinhadas conforme o exemplo.

Observações:

- Entrada: `C:\Lab6\entrada6.txt`
- Saída: `C:\Lab6\seunome6.txt`
- Tamanho da descrição: máximo 70 caracteres.
- Tamanho de cada linha nos cabeçalhos: máximo 70 caracteres.
- Tarefas diferentes podem ter mesmo valor de prioridade, sim.  
Nesse caso, a agenda pode escolher qualquer uma delas.
- A saída deve seguir o **alinhamento** no formato do exemplo.