

(TAD\_gen\_05) Você foi contratado por uma empresa responsável por um sistema operacional para desenvolver um TAD genérico capaz de armazenar, em um vetor de ponteiros, tarefas a serem executadas com determinado nível de prioridade (a tarefa de maior prioridade deve ser executada primeiro). Este TAD representa uma agenda de tarefas a serem executadas. Ele também deve ter uma função para percorrer todas as tarefas armazenadas, executando-as na ordem de prioridade. Ao terminar a execução da tarefa, ela deve ser apagada (memória desalocada).

Este TAD deve ter a capacidade de processar qualquer tipo de tarefa, incluindo algumas não previstas por você. Assim, cada tarefa deve ser cadastrada acompanhada de dois *callbacks*, um que *será* executado na função de percorrer o vetor de tarefas, e outro usado para liberar a memória alocada.

Para testar o TAD implementado, faça um programa que leia 3 tipos de tarefas diferentes:

- Impressão de mensagem de texto na tela; (Prefixo I)
- Realizar uma soma de dois números, imprimindo o resultado; (Prefixo S)
- Realizar uma multiplicação de dois números, imprimindo o resultado. (Prefixo M)

Você deverá ler uma lista de tarefas destes 3 tipos (caso não tenha um dos prefixos, , armazenar na agenda de tarefas e executá-las em ordem de prioridade (maior para o menor).

Definição dos formatos de entrada e saída:

**Entrada:** Primeiro, seu programa deve receber a quantidade de tarefas a serem executadas. Após, deve receber as tarefas (uma por linha), no seguinte formato:

```
X I "mensagem_a_ser_impressa"  
X S N1 N2  
X M N1 N2
```

Onde X é um inteiro definindo a prioridade da tarefa (quanto maior, mais urgência).

**Saída:** Após o recebimento de todas as tarefas, você deve ordená-las e executá-las.

Ver exemplos de formato de entrada e saída nos arquivos fornecidos com a questão.