

Aula prática 4

A aula prática 4 tem o objetivo de exercitar os conhecimentos sobre filas e pilhas.

Nesta aula serão, também, consolidados os conhecimentos sobre ordenação de listas e/ou vetores.

1 – O armazém de um porto marítimo guarda contentores à espera de serem carregados em navios que os irão transportar aos seus destinos. Cada contentor tem informação sobre o destino e o valor total das mercadorias dentro dele. Ao chegarem ao armazém, os contentores são descarregados dos camiões e empilhados o mais próximo possível do local de carga dos navios, como ilustra a figura seguinte.



O armazém possui uma capacidade que é definida pela altura de cada pilha, que limita o número de contentores que podem ser postos uns sobre os outros, e pelo comprimento da fila de pilhas de contentores. Novos contentores são empilhados de um lado do armazém (entrada) e são retirados para os navios do outro lado (saída). À medida que as pilhas vão atingindo a sua capacidade, na entrada, são deslocadas para o lado da saída por um tapete rolante automático.

Para facilitar o desempenho na aula prática são fornecidos ficheiros .c (e respetivo ficheiro .h) com algumas estruturas definidas e código parcialmente implementado:

- `contentor` – estrutura do contentor e código associado
- `armazem` – estrutura do armazém e código associado
- `lista_contentores` – estrutura e código para uma lista de contentores
- `pilha_contentores` – estrutura e código para uma pilha de contentores
- `fila_contentores` – estrutura e código para uma fila de pilhas de contentores
- `teste_armazem` – ficheiro de teste para o código a desenvolver nesta aula.

Nos ficheiros auxiliares as implementações de fila e de pilha utilizam listas ligadas.

Complete as funções pedidas de modo a desenvolver uma aplicação de gestão do armazém:

a) Implemente as funções

```
contentor* contentor_novo(char* dest, float val);  
void contentor_apaga(contentor* contr);
```

do ficheiro `contentor.c` de modo a utilizar memória dinâmica na representação dos contentores.

- b) Analise a estrutura de dados `armazem` no ficheiro `armazem.h` – contém uma fila de pilhas de contentores e uma lista (historico) onde são guardados todos os contentores que passaram pelo armazém. Considere que a capacidade do armazém é de 4 pilhas enfileiradas com a altura máxima de 3 contentores. Tendo em conta esta estrutura, implemente a seguinte função no ficheiro `armazem.c`

```
armazem* armazem_novo();
```

que devolve um apontador para um armazém.

- c) Implemente a seguinte função no ficheiro `armazem.c`

```
int armazem_vazio(armazem* arm);
```

que retorna 1 (um) se o armazém não tiver contentores, e 0 (zero) caso contrário;

- d) Implemente a seguinte função no ficheiro `armazem.c`

```
int armazem_cheio(armazem* arm);
```

que retorna 1 (um) quando a capacidade do armazém para receber novos contentores se esgotou, ou seja, quando a pilha mais à direita está cheia, e 0 (zero) caso contrário.

- e) Implemente a seguinte função no ficheiro `armazem.c`

```
int descarregar_camiao(armazem* arm, contentor* contr);
```

responsável por empilhar os contentores no armazém. Quando o número máximo de contentores numa pilha é atingido, começa-se a empilhá-los noutra pilha. Quando o número de pilhas atinge o comprimento máximo do armazém, e já não se pode colocar mais contentores, a função deve retornar 0 (zero) assinalando erro, caso contrário deve retornar 1 (um). Cada novo contentor que dá entrada no armazém deve ser guardado na lista de histórico.

- f) Implemente a seguinte função no ficheiro `armazem.c`

```
contentor* carregar_navio(armazem* arm);
```

que retira do armazém o contentor do topo da pilha mais próxima do navio a carregar. A função deve retornar um apontador para o contentor retirado. Quando não há mais contentores no armazém deve ser retornado NULL.

- g) Implemente a seguinte função no ficheiro `armazem.c`

```
void imprimir_registo(armazem* arm);
```

que imprime os registos históricos do armazém.

- h) Todos os dias a companhia de seguros deve avaliar o valor das mercadorias no armazém, somando os valores de todos os contentores lá guardados. Implemente a função seguinte no ficheiro `armazem.c`

```
float referencia_seguro(armazem* arm);
```

que retorna o valor total de mercadorias nos contentores do armazém. Altere estruturas e funções existentes de modo a criar uma solução simples.