

Lista Prática da Unidade II

Estruturas de Dados

1. Escreva um programa que simule o controle de uma pista de decolagem de aviões em um aeroporto. Neste programa, o usuário deve ser capaz de realizar as seguintes tarefas:
 - (a) Listar o número de aviões aguardando na fila de decolagem
 - (b) autorizar a decolagem do primeiro avião da fila
 - (c) adicionar um avião a fila de espera
 - (d) listar os aviões na fila de espera
 - (e) listar as características do primeiro avião da fila
2. Utilizando uma **pilha**, escreva um método que receba um número inteiro positivo no formato decimal e converte este número para o formato binário. Exemplos: $5 \rightarrow 101$; $13 \rightarrow 1101$; $1 \rightarrow 1$. O método retorna uma lista encadeada com os dígitos que fazem parte do número em binário.
3. Suponha que um dado problema requer o uso de duas pilhas, onde cada pilha suporta no máximo 50 elementos e em nenhum momento as duas pilhas terão juntas mais do que 80 elementos. Assim, é possível implementar as duas pilhas em um único vetor usando apenas 80 posições ao invés de 100. Implemente a estrutura de dados e as de empilhar e desempilhar para estas duas pilhas.
4.) Escreva um programa solicite ao usuário uma sequência de caracteres sem limite de máximo de tamanho e realize as seguintes operações usando uma pilha:
 - (a) Imprimir o texto na ordem inversa;
 - (b) Verificar se o texto é um palíndromo, ou seja, se a string é escrita da mesma maneira de frente para trás e de trás para frente. Ignore espaços e pontos.
5. Considere uma pilha P vazia e uma fila F não vazia. Utilizando apenas os testes de fila e pilha vazias, as operações **Enfileira**, **Desenfileira**, **Empilha**, **Desempilha**, e uma variável **aux** do **TipoItem**, escreva uma função que inverta a ordem dos elementos da fila.
6. Existem partes de sistemas operacionais que cuidam da ordem em que os programas devem ser executados. Por exemplo, em um sistema de computação de tempo compartilhado (“time-shared”) existe a necessidade de manter um conjunto de processos em uma fila, esperando para serem executados. Escreva um programa que seja capaz de ler uma série de solicitações para (Assuma que cada processo é representado por um registro composto por um número identificador do processo) :
 - (a) Incluir novos processos na fila de processo;
 - (b) Retirar da fila o processo com o maior tempo de espera;
 - (c) Imprimir o conteúdo da lista de processo em determinado momento.
7. Escreva um método para inverter a ordem dos elementos de uma **fila**, usando uma **pilha** como estrutura auxiliar.

Referências

https://www.ic.unicamp.br/~ra144681/courses/mc202/slides/files/Lista02-Listas_Pilhas_Filas.pdf
<http://www.decom.ufop.br/menotti/edI101/listas/lista-Fila.pdf>