

Exercícios – Filas

1. Faça um programa para a manipulação de filas com alocação seqüencial e estática na memória do computador. Crie as seguintes rotinas genéricas de manipulação de dados:

Definições elementares:

Constante

Tamanho = 10;

Tipo

Tipo_Dado = Integer;

Tipo_Fila = Record

Dado: array [1..Tamanho] of Tipo_Dado;

Inicio, Fim: Integer;

fim;

Rotinas:

Procedure Inicializa_Fila (Var F: Tipo_Fila);

Function Insere_Fila (Var F: Tipo_Fila; Dado: Tipo_Dado): Boolean;

Function Retira_Fila (Var F: Tipo_Fila; Var Dado: Tipo_Dado): Boolean;

Function Fila_Vazia (F: Tipo_Fila): Boolean; { Testa para ver se a fila está vazia }

Function Quantidade_Fila (F: Tipo_Fila): Integer; { Indica quantos dados tem na fila }

Procedure Lista_Fila (Var F: Tipo_Fila); { Retira os dados da fila para mostra-los }

Procedure Exibe_Fila (F: Tipo_Fila); { Exibe os dados, mantendo a fila inalterada }

Procedure Inverte_Fila (Var F: Tipo_Fila); { Inverte a ordem dos dados da fila }

Procedure Esvazia_Fila (Var F: Tipo_Fila); { Retira todos os dados contidos na fila }

2. Construa uma função retire_cinco(p) que retire os 5 primeiros elementos da fila f e devolva um valor lógico que indica se foi possível concluir a operação sem que a fila fique vazia.

3. Construa uma função que determine o número de elementos existentes numa fila f.

4. Desenvolva uma função ultimo(f) que devolva o último elemento da fila f.

5. Escreva um procedimento que recebe duas filas, que contém valores numéricos ordenados. O procedimento deverá formar uma terceira fila, também ordenada, na qual estarão os valores armazenados nas filas originais.

6. Considere:

a) uma pilha implementada sobre um arranjo (vetor) P(20) a partir da primeira posição. O topo da pilha está armazenado na variável TOPO;

b) uma fila implementada sobre outro arranjo F(20), a partir da posição 1. O início e o final da fila estão armazenados, respectivamente, nas variáveis INICIO e FIM. É utilizada a otimização de espaço do arranjo (quando a fila chegar ao final do arranjo, se ainda existir espaço disponível no início do arranjo, este será utilizado).

Escreva:

- a) um procedimento que retire um elemento do topo da pilha e o coloque na fila;
- b) um procedimento que retire um elemento da fila e o coloque na pilha.

Exercícios - Pilhas

1. Faça um programa para a manipulação de pilhas com alocação seqüencial e estática na memória do computador. Crie as seguintes rotinas genéricas de manipulação de dados:

Definições elementares:

Const

Tamanho = 10;

Type

Tipo_Dado = Integer;

Tipo_Pilha = Record

Dado: array [1..Tamanho] of Tipo_Dado;

Base,Topo: Integer;

End;

Rotinas:

Procedure Inicializa_Pilha (Var P: Tipo_Pilha);

Function Insere_Pilha (Var P: Tipo_Pilha; Dado: Tipo_Dado): Boolean;

Function Retira_Pilha (Var P: Tipo_Pilha; Var Dado: Tipo_Dado): Boolean;

Function Pilha_Vazia (P: Tipo_Pilha): Boolean; { Testa para ver se a pilha está vazia }

Function Quantidade_Pilha (P: Tipo_Pilha): Integer; { Indica quantos dados tem na pilha }

Procedure Lista_Pilha (Var P: Tipo_Pilha); { Desempilha para listar os dados }

Procedure Exibe_Pilha (P: Tipo_Pilha); { Exibe os dados, mantendo a pilha inalterada }

Procedure Inverte_Pilha (Var P: Tipo_Pilha); { Inverte a ordem dos dados da pilha }

Procedure Esvazia_Pilha (Var P: Tipo_Pilha); { Desempilha todos os dados da pilha }

2. Construa uma função elemento(p,n) que devolva o elemento da pilha colocado na posição n, a contar do topo.
3. Construa uma função pop_cinco(p) que retire os 5 primeiros elementos da pilha p e devolva um valor lógico que indica se foi possível concluir a operação sem que a pilha fique vazia.
4. Construa um procedimento inverte(x,n), que, com o auxílio de uma pilha, inverte um vetor x que contém n números reais.
5. Dada uma pilha construa uma subrotina que altere o elemento situado na posição n (a contar do topo) para o valor x. No caso de a pilha ter menos de n elementos deve ser alterado o último. Os restantes elementos da pilha não devem ser alterados, apesar de poderem ser movimentados. No caso de ser necessário pode ser utilizada uma estrutura auxiliar.
6. Faça um procedimento que leia um conjunto de valores inteiros e armazene estes valores em duas pilhas, uma para os valores positivos lidos e a outra, para os valores negativos. As pilhas devem ser implementadas sobre um mesmo arranjo PILHAS, a partir de posições fornecidas como parâmetros. Em caso de overflow de alguma das pilhas, deve ser emitida uma mensagem.
7. Considere um arranjo unidimensional utilizado para implementar duas pilhas. As bases das pilhas estarão uma em cada extremidade do arranjo. O arranjo possui N elementos, inteiros. Cada uma das pilhas apresenta um índice de topo de pilha, os quais são nulos quando a pilha é vazia. Escreva:

- c) um procedimento para incluir um elemento no topo da pilha B, atualizando o indicador de topo desta pilha. Este procedimento receberá, como parâmetros, o valor do elemento a ser incluído na pilha B, os índices correspondentes ao topo das duas pilhas e o número de elementos do arranjo. O procedimento deverá testar se existe uma célula livre para incluir o elemento (se as duas pilhas não estão ocupando todo o arranjo) e, caso isto ocorrer, não efetuar a inclusão e enviar uma mensagem avisando;
 - d) um procedimento que calcule a quantidade de elementos da pilha A que também aparecem na pilha B. Elementos repetidos na mesma pilha devem ser contados apenas uma vez.
8. Escreva uma subrotina em Java que, utilizando a estrutura de pilha (utilizar armazenamento estático - vetor), receba uma palavra ou frase e apresente-a em ordem inversa.
9. Reescrever o exercício 8 utilizando alocação dinâmica de memória.
- 10.

2. A conversão de um valor decimal para o seu correspondente em binário é feita pelas sucessivas divisões dele por 2 até que o quociente seja 0. O representante binário desse número será composto por todos os restos, mas na ordem inversa à que foram calculados. Elabore um algoritmo e um programa capazes de resolver essa questão utilizando o conceito de pilhas. O número deverá ser fornecido pelo usuário.
3. Construa um algoritmo/programa que administre as filas de reservas de filmes em uma videolocadora, levando em conta que para cada filme existem sete filas — uma para cada dia da semana — e é o usuário quem determina qual é o dia da semana de sua preferência para alugar o filme. O cliente é informado da disponibilidade da fita, e quando é confirmada, a sua locação ele deve sair da fila. O número de cópias de cada fita não deverá ser considerado nessa solução, devendo ser considerada uma fita de cada filme.



Calculadora pós-fixada

Um bom exemplo de aplicação de pilha é o funcionamento das calculadoras da HP (Hewlett-Packard). Elas trabalham com expressões pós-fixadas, então para se avaliar uma expressão como $(1-2)*(4+5)$, a mesma pode ser digitada $1\ 2\ -\ 4\ 5\ +\ *$.

O funcionamento dessas calculadoras é muito simples. Cada operando é empilhado numa pilha de valores. Quando se encontra um operador, desempilha-se o número apropriado de operandos (dois para operadores binários e um para operadores unários), realiza-se a operação devida e empilha-se o resultado. Deste modo, na expressão acima, são empilhados os valores 1 e 2. Quando aparece o operador -, 1 e 2 são desempilhados e o resultado da operação, no caso -1 ($= 1 - 2$), é colocado no topo da pilha. A seguir, 4 e 5 são empilhados. O operador seguinte, +, desempilha o 4 e o 5 e empilha o resultado **da soma, 9**. **Nessa hora**, estão na pilha os dois resultados parciais, -1 na base e 9 no topo. O operador *, então, desempilha os dois e coloca -9 ($= -1 * 9$) no topo da pilha.

Exemplo.:

Expressão = $(1-2)*(4+5)$

Notação Polonesa (pós-fixa) = $1\ 2\ -\ 4\ 5\ +\ *$

String exp = "1 2 - 4 5 + *";

Passo 1	Passo 2	Passo 3	Passo 4	Passo 5	Passo 6	Passo 7
1 2 - 4 5 + * ^	1 2 - 4 5 + * ^	1 2 - 4 5 + * ^	1 2 - 4 5 + * ^	1 2 - 4 5 + * ^	1 2 - 4 5 + * ^	1 2 - 4 5 + * ^
Push(1)	push(2)	n1 = pop() n2 = pop() push(n1-n2)	push(4)	push(5)	n1 = pop() n2 = pop() push(n1+n2)	n1 = pop() n2 = pop() push(n1*n2)
<div> <div></div> <div></div> <div></div> <div></div> <div>1 ← topo</div> </div>	<div> <div></div> <div></div> <div></div> <div>2 ← topo</div> <div>1 ← topo</div> </div>	<div> <div></div> <div></div> <div></div> <div></div> <div>-1 ← topo</div> </div>	<div> <div></div> <div></div> <div></div> <div>4 ← topo</div> <div>-1 ← topo</div> </div>	<div> <div></div> <div></div> <div>5 ← topo</div> <div>4 ← topo</div> <div>-1 ← topo</div> </div>	<div> <div></div> <div></div> <div></div> <div>9 ← topo</div> <div>-1 ← topo</div> </div>	<div> <div></div> <div></div> <div></div> <div></div> <div>-9 ← topo</div> </div>

Considere que a String contendo a expressão é acessada caractere a caractere, e o caractere acessado a cada passo é indicado pela seta (^). Abaixo de cada passo são mostradas as operações. Considere-as e as implemente uma aplicação para a calculadora usando o TAD Pilha definido em Java.

Agência Bancária

Em uma agência bancária dos anos 80 existem 4 caixas e uma fila para cada caixa. Implementar um programa para simulação das filas na agência. Considerar que o tamanho das filas é indeterminado.

Nessa agência está sendo realizada uma pesquisa e todos as pessoas ao serem atendidas na entrada da fila eles preenchem uma ficha fornecendo as seguintes informações: nome e a operação a ser realizada (depósito ou saque) e informação se é cliente do banco ou não.

Para o programa use um vetor de 4 posições para representar as Filas, e em cada posição do vetor use uma Fila para representar as pessoas pertencentes a elas. Para simular a entrada de pessoas nessa fila use o objeto **Math.random()**, que escolhe de forma aleatória as operações que serão realizadas, se chega alguém no banco e em qual fila ele irá entrar, ou se alguém já foi atendido em algum dos caixas.

