

ARA7125 Estruturas de Dados I
Primeiro Semestre - 2015
Segunda Lista de Exercícios

1. – Implemente uma fila em uma lista encadeada circular com cabeça (faça funções que implementam as operações de Inserção e Remoção). O primeiro elemento da fila ficará na segunda célula e o último elemento ficará na célula anterior à cabeça. A função de Inserção deve receber um (e somente um) ponteiro para a cabeça da lista, e o conteúdo a ser inserido na fila. A função Remoção deve receber um (e somente um) ponteiro que aponta para a cabeça lista. Pergunta: É possível manipular essa fila? Como inserir, se temos acesso somente à cabeça da lista? Como remover, se temos acesso somente à cabeça da lista?

2. – Faça uma função com o seguinte protótipo

`int Verifica_Operador_Operando (char op []);`

A função recebe uma cadeia de caracteres `op` e devolve 1 se `op` é um operador, ou devolve 0 se `op` é um operando. Use a função `strcmp` da biblioteca `string.h`

3. – (Aplicação para pilha) Escreva um programa que calcula o valor de uma expressão na notação posfixa terminada por um `'.'` (operandos, operadores e `'.'` separados por um espaço). Exemplos:

- Se o seu programa receber `5 6 7 * + .`, então o resultado é 47, pois $(5 + 6 * 7) = 47$.
- Se o seu programa receber `6 9 * 3 60 - 4 / + .`, então o resultado é 39.75, pois $(6 * 9 + (3 - 60) / 4) = 39.75$.

Use a função do exercício anterior na solução deste exercício.

4. – Escreva uma função que recebe uma sequência de caracteres e usa uma pilha para devolver a sequência invertida. Por exemplo, se a função receber “banana” deve usar uma pilha para devolver “ananab”.

5. – (Aplicação para pilha) Escreva uma função que recebe uma frase f (ou seja, uma sequência de caracteres) e devolve 1, se f é um palíndromo. Uma frase é um palíndromo se ela pode ser lida da esquerda para à direita ou da direita para à esquerda. Por exemplo, “O galo ama o lago.”, “Socorram-me, subi no ônibus em Marrocos!”, “A cera causa sua careca.”, “A diva em argel alegra-me a vida.”.

6. – (PF) (Aplicação para fila) Imagine um tabuleiro quadrado n por n . As casas “livres” são marcadas com 0 e as casas “bloqueadas” com -1. As casas $(1, 1)$ e (n, n) estão livres. Faça um programa que ajuda uma formiga que está na casa $(1, 1)$ a chegar à casa (n, n) . Uma restrição é que a formiga só pode se deslocar para uma casa livre que esteja à direita, à esquerda, acima ou abaixo da casa em que está. Exemplos de entrada e saída para o seu programa são dados em seguida.

Entrada:

```

8
0  -1  -1  -1  -1  -1  -1  -1
0  -1   0   0   0  -1  -1  -1
0  -1   0  -1   0  -1   0  -1
0  -1  -1  -1   0  -1   0  -1
-1   0   0   0   0  -1   0  -1
-1   0  -1  -1  -1  -1   0  -1
-1   0   0   0   0   0   0   0
-1   0  -1  -1  -1  -1  -1   0

```

Saída:

Não existe um caminho de $(1,1)$ até $(8,8)$.

Entrada:

```

9
0   0   0   0   0   0   0  -1  -1
-1  -1   0   0  -1  -1  -1  -1  -1
0  -1   0  -1   0  -1  -1  -1  -1
-1  -1  -1   0   0  -1   0  -1  -1
-1   0   0   0   0  -1   0  -1  -1
-1   0  -1  -1   0  -1   0  -1  -1
-1   0   0   0   0   0   0   0  -1
-1   0  -1  -1  -1  -1  -1   0   0
-1   0  -1  -1   0  -1   0  -1   0

```

Saída:

Não existe um caminho de $(1,1)$ até $(9,9)$.

Entrada:

10

0	0	0	0	0	0	0	-1	-1	0
-1	-1	0	0	-1	-1	0	-1	-1	0
0	0	0	-1	0	-1	0	-1	-1	0
-1	-1	-1	0	0	-1	0	-1	-1	0
-1	0	0	0	0	-1	0	-1	-1	0
-1	0	-1	-1	0	-1	0	-1	-1	0
-1	0	0	0	0	0	0	0	-1	0
-1	0	-1	-1	-1	0	-1	0	0	0
-1	0	-1	-1	-1	0	-1	0	0	0
-1	0	0	0	0	0	0	-1	0	0

Saída:

Existe um caminho de $(1,1)$ até $(10,10)$.