

ESTRUTURA DE DADOS: PILHAS

Prof. Juliana Kaizer Vizzotto

DLSC/CT/UFSM

2015a

1 Trabalho I

O objetivo deste trabalho ¹ é, utilizando apenas as operações sobre o TAD Pilha de inteiros (alocar, desalocar, push, pop e numeroElementos) e recursividade, escrever funções para inserir e remover elementos em qualquer posição da pilha. Ou seja, a pilha será usada para “simular” operações de inserção e remoção em vetor.

Neste trabalho, você deve:

1. Definir um TAD pilha armazenado em um vetor alocado dinamicamente;
 - (a) Alocar a memória da pilha (criar);
 - (b) Desalocar a memória da pilha (destruir);
 - (c) Inserir um elemento no topo da pilha (push);
 - (d) Remover um elemento do topo da pilha (pop);
 - (e) Retornar o número de elementos da pilha (numeroElementos);
2. Escrever funções para:
 - (a) Inserir um elemento na i-ésima posição da pilha (utilizando recursividade);
 - (b) Remover um elemento da i-ésima posição da pilha (utilizando recursividade);

Considere o seguinte o exemplo de entrada (lido da entrada padrão):

```
p 1
p 2
i 3 4
q
q
i 1 5
```

¹Trabalho adaptado do icmc/USP

```

i 1 6
r 0
r 1
q
q
x

```

Cada linha representa um tipo de comando que o programa deve interpretar da seguinte maneira:

- “p <n>” : push = insere o número jn_i no topo da pilha;
- “q” : pop = remove o número do topo da pilha e apresenta o seu valor na tela;
- “i <pos> <n>” = insere o número jn_i na posição $jpos_i$ da pilha (começando do zero, a partir da base da pilha). Completa a pilha com zeros até chegar na posição desejada.
- “r <pos>” = remove o número jn_i da posição $jpos_i$ da pilha e apresenta o valor na tela;
- “x” = sair do sistema.

A saída do programa deve ser apresentada na saída padrão, consistindo de várias linhas de acordo com os comandos de entrada:

- “p <n>”: push = não apresenta nada na tela (nem quebra de linha) ;

```
printf ("%d\n",valor);
```
- “q” : pop = remove o número do topo da pilha e apresenta o valor na tela;
- “i <pos> <n>” = apresenta o número de operações push executadas para inserir o valor nessa posição;

```
printf ("%d\n",numerodePushs);
```
- “r <pos>” = remove o número jn_i da posição $jpos_i$ da pilha e apresenta o valor na tela e o número de operações push executadas para fazer isso.

```
printf ("%d%d\n",valor,numerodePushs);
```

- “x” = desalocar a pilha e sair do programa.

Para o exemplo de entrada acima, a saída esperada é:

2

4

0

2

3

13

51

2

6