

Algoritmo e Estrutura de Dados I

~~Tarefa Prática 3 – Quinta-Feira~~

Pilhas

Boa tarde campeão, você programou o MergeSort tão bem (#sqn) que sua fama se espalhou pelas galáxias e chegou até as Forças Imperiais.

Darth Vader está adquirindo um novo estoque de equipamentos para os StormTroopers e precisa da sua ajuda para criar uma pilha de equipamentos que vai acelerar a recepção e a entrega aos StormTroopers.

Todo o equipamento que chegar deve ser colocado no topo da pilha, e todo equipamento que sair, deve ser retirado do topo da pilha.

Sua tarefa é completar as funções `push()`, `pop()` e `top()`.

1) A função `push()` deve inserir um novo equipamento no topo da pilha, ex.:

```
printStack ( ) = Pilha Vazia

push (Laser Rifle)
printStack ( ) = Laser Rifle

push (Laser Cannon)
printStack ( ) = Laser Cannon
                  Laser Rifle

push (StormTrooper Helmet)
printStack ( ) = StormTrooper Helmet
                  Laser Cannon
                  Laser Rifle
```

As funções `pop()` e `top()` devem fazer a verificação de pilha vazia.

2) Você deve analisar o case 2 do `switch(option)` na função `main()` e tirar suas próprias conclusões de como implementar o `pop()`. O resultado da função `pop()`, em conjunto com as operações do case 2 do `switch(option)`, deve ser a retirada do item do topo da pilha e sua impressão no terminal (a impressão já está feita no case 2).

Chamada da função	Impressão no Terminal
<code>pop()</code>	StormTrooper Helmet
<code>pop()</code>	Laser Cannon

3) A função `top()` deve retornar o elemento do topo da pilha.

Chamada da função	Impressão no Terminal
<code>top()</code>	Laser Rifle

Boa sorte.

Que o lado negro da força esteja com você.

O código fornecido não pode ser modificado, somente deve ser implementado o que foi pedido e métodos auxiliares, caso seja necessário. Caso seja implementado código fora das funções exigidas, o mesmo deve ser indicado.

A função *main* fornecida a vocês executa uma sequência de testes pré-definida. Porém, nem todas as funcionalidades possíveis são testadas e passar em todos os testes fornecidos não garante nota máxima na avaliação.

Além disso código que não compila receberá nota zero, código que não liberar toda a memória alocada não receberá mais do que metade da nota, mesmo que o programa faça o que foi solicitado. Nota 10 é aquele trabalho que fez o solicitado e o fez de forma eficiente, ou seja, sem processamento e memória desnecessários para completar a tarefa. Além desses parâmetros, quando a resposta dada não estiver correta a avaliação tentará identificar se estava na direção correta para assim atribuir uma nota intermediária que represente o quanto foi feito e o quanto de processamento e memória foi desperdiçado.