

- 1) Faça um método que recebe duas pilhas formadas por valores numéricos ordenados (no topo das pilhas estão os menores valores de cada uma delas). O método deverá montar uma nova pilha na qual estarão os valores das pilhas recebidas, ordenados. Na pilha resultante, entretanto, os maiores valores deverão estar no topo (3,0)

Questão 1 (3,3)

Escreva um algoritmo que **use uma pilha** para *inverter a ordem* das letras de cada palavra de uma *string*, preservando a ordem das palavras. Por exemplo, dado o texto
ESTE EXERCICIO E MUITO FACIL a saída deve ser ETSE OICICREXE E OTIUM
LICAF.

Atenção: NÃO é pedido que se implemente a pilha. Ela já existe. A sua implementação resultará em perda de pontos.

Questão 2 (3,0)

Escreva um algoritmo que **use uma pilha** para *inverter a ordem* das letras de cada palavra de uma *string*, preservando a ordem das palavras. Por exemplo, dado o texto
ESTE EXERCICIO E MUITO FACIL a saída deve ser ETSE OICICREXE E OTIUM
LICAF.

Atenção: NÃO é pedido que se implemente a pilha. Ela já existe. A sua implementação resultará em perda de pontos.

- 1) Faça um método que recebe duas pilhas formadas por valores numéricos ordenados (no topo das pilhas estão os menores valores de cada uma delas). O

método deverá montar uma nova pilha na qual estarão os valores das pilhas recebidas, ordenados. Na pilha resultante, entretanto, os maiores valores deverão estar no topo (2,0)

Questão 1 (POSCOMP 2010) Um estudante de computação precisa resolver um problema bastante importante, que é executar as operações que estão descritas abaixo, cuja estrutura é uma pilha. Tão logo ele retire algum elemento desta pilha, estes deverão ser inseridos em uma fila, cuja entrada é pela esquerda e a saída, pela direita.

Assinale a alternativa que contém a sequência correta de entrada dos elementos na fila.

PUSH P
PUSH E
PUSH R
PUSH T
PUSH O
POP
POP
PUSH S
PUSH O
PUSH L
POP
POP
POP

- a) S - O - L - T - O
- b) O - T - R - E - P
- c) P - E - R - T - O
- d) O - T - L - O - S
- e) P - O - R - L - S

Questão 4 (POSCOMP 2011) As estruturas de dados lineares (fila, pilha e lista) são muito utilizadas para resolver problemas computacionais.

Cada uma dessas estruturas pode ser implementada com diferentes características e atendem a diferentes tipos de problemas.

Sobre as características dessas estruturas de dados, atribua V (verdadeiro) ou F (falso) para as afirmativas a seguir.

- () Em uma pilha, o último elemento a entrar é o primeiro a sair.
- () Em uma fila, o primeiro elemento a entrar é o último a sair.
- () Uma lista permite que as inserções possam ser feitas em qualquer lugar (posição), mas as remoções, não.
- () Em uma lista circular com encadeamento simples, o primeiro elemento aponta para o segundo e para o último.
- () Para remover um elemento de uma lista duplamente encadeada, deve-se alterar o encadeamento dos elementos anterior e próximo ao elemento removido.

Assinale a alternativa que contém, de cima para baixo, a sequência correta.

- a) V, F, V, F, V.
- b) V, F, F, V, F.
- c) V, F, F, F, V.
- d) F, V, V, F, F.
- e) F, F, V, V, V.

Questão 8 (ENADE 2011 - Modificada) No desenvolvimento de um *software* que analisa bases de DNA, representadas pelas letras A, C, G, T, utilizou-se as estruturas de dados: pilha e fila. Considere que, se uma sequência representa uma pilha, o topo é o elemento mais à esquerda; e se uma sequência representa uma fila, a sua frente é o elemento mais à esquerda.

Analise o seguinte cenário: “a sequência inicial ficou armazenada na primeira estrutura de dados na seguinte ordem: (A,G,T,C,A,G,T,T). Cada elemento foi retirado da primeira estrutura de dados e inserido na segunda estrutura de dados, e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A). Finalmente, cada elemento foi retirado da segunda estrutura de dados e inserido na terceira estrutura de dados e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A)”.

Qual a única sequência de estruturas de dados apresentadas a seguir pode ter sido usada no cenário descrito acima?

- a) Fila - Fila - Pilha.
- b) Fila - Pilha - Pilha.
- c) Pilha - Fila - Pilha.
- d) Pilha - Pilha - Fila.
- e) Pilha - Pilha - Pilha.

Para cada questão de 1 a 3, assinale a alternativa correta e **justifique** sua resposta. Só terá validade resposta com justificativa coerente. Cada questão objetiva (de 1 a 3) vale 1,0 ponto.

Questão1: (ENADE 2011) No desenvolvimento de um *software* que analisa bases de DNA, representadas pelas letras A, C, G, T, utilizou-se as estruturas de dados: pilha e fila. Considere que, se uma sequência representa uma pilha, o topo é o elemento mais à esquerda; e se uma sequência representa uma fila, a sua frente é o elemento mais à esquerda.

Analise o seguinte cenário: “a sequência inicial ficou armazenada na primeira estrutura de dados na seguinte ordem: (A,G,T,C,A,G,T,T). Cada elemento foi retirado da primeira estrutura de dados e inserido na segunda estrutura de dados, e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A). Finalmente, cada elemento foi retirado da segunda estrutura de dados e inserido na terceira estrutura de dados e a sequência ficou armazenada na seguinte ordem: (T,T,G,A,C,T,G,A)”.

Qual a única sequência de estruturas de dados apresentadas a seguir pode ter sido usada no cenário descrito acima?

- A) Fila - Pilha - Fila.
- B) Fila - Fila - Pilha.
- C) Fila - Pilha - Pilha.
- D) Pilha - Fila - Pilha.
- E) Pilha - Pilha - Pilha.

1. Use as operações de Pilha (criar, push, pop, topo e estah_vazia) para construir rotinas que façam o seguinte:
 - a) Dado um inteiro n, apresentar o n-ésimo elemento a partir do topo da pilha, deixando a pilha sem seus n elementos superiores (1,0)
 - b) Dado um inteiro n, apresentar o n-ésimo elemento a partir do topo da pilha, deixando a pilha inalterada (1,5)

3 Escreva um algoritmo, **usando as funções do TAD pilha** (criar, push, pop, topo, estahVazia), que inverta as letras de cada palavra de um texto terminado por “.”, preservando a ordem das palavras. Por exemplo, dado o texto:

este exercício é muito fácil.

a saída deve ser

etseoicícrexe é otiumlicáf.

A saída deve ser apresentada na tela.

Valor da questão (1,5)

1. Dadas as seguintes operações sobre Pilhas de caracteres, independente de sua implementação, resolva a questão abaixo (2,5).

```
void criar (pilha *p);  
void push(pilha *p, char valor);  
void pop(pilha *p);  
char topo (pilha p);  
int estahVazia (pilha p);
```

Examine uma expressão matemática que possui vários conjuntos de parênteses agrupados. Por exemplo:

$$7 - ((X * ((X + Y) / (J - 3)) + Y) / (4 - 2 * 5))$$

Queremos garantir que os parênteses estejam corretamente agrupados, ou seja, desejamos verificar se:

1. Existe um número igual de parênteses esquerdos e direitos.
2. Todo o parêntese da direita está precedido por um parêntese da esquerda correspondente

Expressões como

$((A + B))$ ou $A + B ($

violam o critério 1, e expressões como:

$)A + B (-C$ ou $(A + B)) - (C + D$

violam o critério 2.

Para solucionar esse problema, imagine cada parêntese da esquerda como uma abertura de um escopo. A profundidade do aninhamento (ou profundidade do agrupamento) em determinado ponto numa expressão é o número de escopos abertos, mas ainda não fechados nesse ponto. Isso corresponderá ao número de parênteses da esquerda encontrados cujos correspondentes parênteses da direita ainda não foram encontrados. Determinemos a contagem de parênteses em determinado ponto numa expressão como o número de parênteses da esquerda menos o número de parênteses da direita encontrados ao rastrear a expressão a partir de sua extremidade esquerda até o ponto em questão. Se a contagem de parênteses for não-negativa, ela equivalerá a profundidade do aninhamento. As duas condições que devem vigorar caso os parênteses de uma expressão formem um padrão admissível são as seguintes:

1. A contagem de parênteses no final da expressão é 0. Isso significa que nenhum escopo ficou aberto ou que foi encontrada a mesma quantidade de parênteses da direita e da esquerda.
2. A contagem de parênteses em cada ponto na expressão é não-negativa. Isso implica que não foi encontrado um parêntese da direita para o qual não exista um correspondente parêntese da esquerda

Implemente um programa que, utilizando pilhas, informe se uma determinada expressão forma um padrão de parentização admissível ou não.

Questão 2 (valor 3,0)

Escreva um algoritmo que **use** uma pilha para *inverter a ordem* das letras de cada palavra de uma string, preservando a ordem das palavras. Por exemplo, dado o texto `ESTE EXERCICIO E MUITO FACIL` a saída deve ser `ETSE OICICREXE E OTIUM LICAFL`.

Atenção: NÃO é pedido que se implemente a pilha. Ela já existe. A sua implementação resultará em perde de pontos.

Pode utilizar como condição de fim de string $\rightarrow \text{str}[i] != \$$ (entendendo \$ como fim da string)