



ATIVIDADE AVALIATIVA I – 1º BIMESTRE – VALOR: 3,0 PONTOS – INDIVIDUAL OU DUPLA

1. Simulação de Torneio de Pedra-Papel-Tesoura com Fila

Desenvolva uma simulação de um torneio de Pedra-Papel-Tesoura onde os jogadores competem em pares e os vencedores continuam para a próxima rodada. Utilize uma fila para organizar os jogadores. A cada rodada, dois jogadores são removidos da fila e competem entre si. O vencedor da partida deve retornar ao final da fila, enquanto o perdedor é eliminado do torneio. O processo continua até que reste apenas um jogador na fila, que será o campeão do torneio. Por exemplo, se a fila inicial tiver 8 jogadores, o algoritmo deve continuar removendo pares de jogadores, competindo, e reordenando os vencedores até que reste apenas um vencedor final.

2. Simulação do Funcionamento de um Elevador em um Prédio

Crie uma simulação para o funcionamento de um elevador em um prédio, onde as pessoas entram no elevador e solicitam andares específicos. Use uma fila para armazenar as solicitações de andares. O elevador processa cada solicitação na ordem em que foi recebida, "movendo-se" para o andar desejado e então removendo a solicitação da fila. Se houver múltiplos elevadores, cada um terá sua própria fila de solicitações a ser processada de forma independente.

Por exemplo, se o elevador receber solicitações para os andares 5, 3 e 8, ele deve mover-se para o andar 5, depois para o andar 3, e por último para o andar 8. Novas solicitações devem ser adicionadas ao final da fila do elevador.

3. Simulação do Jogo Torre de Hanói

Implemente uma simulação do jogo Torre de Hanói, onde o objetivo é mover uma pilha de discos de um pino para outro seguindo regras específicas: apenas um disco pode ser movido por vez, e um disco maior não pode ser colocado sobre um disco menor. Utilize pilhas para representar cada um dos três pinos, onde os discos são empilhados do maior (na base) ao menor (no topo). Comece com todos os discos no pino inicial e mova-os para o pino de destino, utilizando o pino auxiliar, se necessário. O jogo deve respeitar a regra de que um disco maior não pode ser colocado sobre um disco menor.

Por exemplo, se o pino inicial contém os discos [3, 2, 1], o objetivo é movê-los para o pino de destino, utilizando o pino auxiliar conforme as regras do jogo.

4. Conversão de Expressão Aritmética da Notação Infixa para a Notação Pós-Fixa

Desenvolva um algoritmo para converter uma expressão aritmética da notação infix (onde os operadores são colocados entre os operandos) para a notação pós-fix (também conhecida como notação polonesa reversa - RPN), onde os operadores são colocados após seus operandos. O programa deve receber uma expressão aritmética em notação infix. A expressão pode conter os operadores +, -, *, /, e ^ (exponenciação), além de parênteses (e). A expressão pode incluir variáveis (representadas por letras) ou números (opcionais).

Exemplos de entrada incluem $A + B * (C ^ D - E) ^ (F + G * H) - I$, $3 + 4 * 2 / (1 - 5) ^ 2 ^ 3$, $A * (B + C) / D$, e $(A + B) * (C + D)$. A saída deve ser a expressão correspondente em notação pós-fix: $A B C D ^ E - F G H * + ^ * + I -$, $3 4 2 * 1 5 - 2 3 ^ ^ / +$, $A B C + * D /$, e $A B + C D + *$, respectivamente.



Utilize uma estrutura de dados **pilha** para armazenar operadores e gerenciar sua precedência. A precedência dos operadores é a seguinte: ^ tem a maior precedência, seguido por * e /, e depois por + e -. A associatividade dos operadores é da esquerda para a direita, exceto para ^, que é da direita para a esquerda. Os parênteses na notação infixa definem a ordem das operações, mas não devem aparecer na saída pós-fixa. Seu código será avaliado pela correção da conversão, pela eficiência do algoritmo e pela clareza da implementação. Certifique-se de incluir comentários explicativos no código.

Avaliação:

As atividades serão avaliadas seguindo o seguinte critério:

- Clareza de código e documentação de cada método. (Ex.: Nomes de variáveis amigáveis, comentários de código e demais boas práticas) **(1,0 PONTO) – (0,25 PONTO POR PROBLEMA)**
- Conclusão de todos os requisitos de cada estrutura. **(1,0 PONTO) - (0,25 PONTO POR PROBLEMA)**
- Entrega das demandas no prazo. **(1,0 PONTO) - (0,25 PONTO POR PROBLEMA)**

Observações importantes:

- A atividade deverá ser **exclusivamente entregue via Blog**, por apenas um dos componentes, contendo os arquivos de código separados por problema e com a primeira linha com comentário contendo o nome dos componentes, os mesmos devem ser compactados com o nome dos alunos, como por exemplo: **“JOAO_DAS_COUVES_MARIA_DAS_CANDONGAS.zip”** e **enviadas via blog exclusivamente em formato .zip**.
- Detectado plágio de atividade ou não conformidade com as estruturas solicitadas na atividade, o trabalho será passível de ter sua pontuação zerada.
- Atividades com 3 ou mais integrantes serão automaticamente zeradas.

DATA MÁXIMA DA ENTREGA: 22-09-2024 – 23H59 – VIA BLOG.