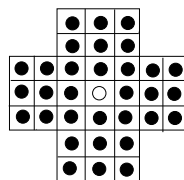




Resta um

1 Descrição

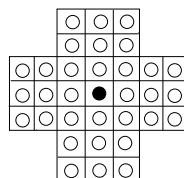
O objetivo deste trabalho é implementar uma técnica de backtrack para resolver o problema de um jogo de tabuleiro conhecido como **Resta um**. No início do jogo, temos 33 buracos e 32 pinos dispostos em forma de cruz, com a posição central vazia.



Ao executar um movimento válido na horizontal ou na vertical, os pinos serão gradualmente removidos do tabuleiro. Um **movimento válido** é descrito da seguinte maneira. Sejam três posições (A, B, C) consecutivas no tabuleiro (na horizontal ou na vertical), com A e B preenchidas e C vazia. O movimento válido consiste em eliminar a peça que está na posição B e mover a peça que está em A para a posição vazia C .



O objetivo do jogo é, após 31 movimentos válidos, obter apenas o pino central na posição central do tabuleiro (aquela que estava vazia no início). Seu programa deve, no final, imprimir uma sequência de movimentos que leva à configuração final desejada.



2 O que entregar

- O código-fonte em linguagem C;
- Listagem do código-fonte;



3 Observações

- Todos os fontes devem necessariamente conter os nomes dos integrantes do grupo.
- Pode ser que seu programa fique durante algum tempo sem responder alguma coisa, dando a impressão de ter entrado em *loop*. Utilize algum artifício para mostrar ao usuário que o seu programa não está travado.
- Anexo a este enunciado encontra-se o arquivo “restaum.out” que é um exemplo de saída do programa.

4 Avaliação

Os critérios de avaliações são os seguintes:

- Atendimento às especificações, funcionamento correto e manipulação adequada de erros;
- Implementação coerente com o objetivo do programa, soluções práticas para resolver o problema proposto, uso correto de algoritmo e estrutura de dados;

A nota da atividade é calculada de acordo com os critérios da tabela a seguir.

Item avaliado	Pontuação máxima
Carregar dados de um arquivo.	até 1,0 ponto
Realizar um movimento válido.	até 1,5 ponto
Estrutura do algoritmo de Backtracking correta.	até 1,5 ponto
Mostrar ao usuário que o programa não está travado.	até 1,5 ponto
No final, apenas o pino central na posição central do tabuleiro.	até 1,0 ponto
Salvar dados para um arquivo.	até 1,0 ponto
Clareza do código-fonte (identação, nomenclatura adequada de variáveis e funções bem distribuídas, comentários pertinentes, etc.).	até 1,0 ponto
Vídeo com a explicação sobre a solução do problema, apresentação do código e sua execução (de 5 a 7 min).	até 2,0 pontos



A tabela a seguir contém critérios de avaliação que podem reduzir a nota final da atividade.

Item avaliado	Redução da nota
Projeto é cópia de outro.	O projeto recebe nota zero.
Projeto não compila e não gera executável.	O projeto recebe nota zero.
O programa exige algum arquivo adicional que não está presente.	-2,5 pontos
Há erros de compilação e/ou o programa trava durante a execução. ¹	-1,0 ponto
Não há identificação do grupo no código-fonte.	-1,0 ponto
Arquivos enviados em formatos incorretos.	-1,0 ponto
Arquivos enviados em formatos incorretos.	-1,0 ponto
Arquivos e/ou pastas intermediárias que são criadas no processo de compilação foram enviadas junto com o código-fonte.	-1,0 ponto

5 Desonestidade acadêmica

Este trabalho deve ser feito individualmente ou em dupla. Evidentemente você pode “discutir” o material dado em classe, inclusive sobre os problemas e “dicas” para chegar às soluções, mas você deve ser responsável pela solução final e pelo desenvolvimento de seus programas. Portanto, cópias totais/parciais dos programas terão nota zero. Além disso, não serão aceitos projetos atrasados. Os casos de desonestidade serão tratados com rigor.