Dredd - Juiz Online

Minutos Restantes: 1960

Usuário: Gabriel Nathan Almeida Silva

Notas:
Q1: 100
Q2: ?
Q3: 100
Total: 67

IAlg - Terceira Lista Avaliativa

Prova Aberta Até: 29/11/2019 23:59:59

Número Máximo de Tentativas: 4

Atenuação da Nota por Tentativa: 0%

Instruções para a prova: A prova é individual. Desligue seu celular. Não converse com os colegas. Não fique olhando para a tela dos colegas.

Questão 1: Ordenação - Corrigir o Merge Sort

Você deverá fazer um programa que recebe um vetor de inteiros V de N posições e o ordena. Para ordenar este vetor, você deverá usar o método **Merge Sort**.

Dica: Abaixo temos uma implementação **incorreta** do Merge Sort. Você deverá corrigí-la (e não refazer o método).

```
void Troca(int &A, int &B)
{
        int aux = A;
        A = B;
        B = aux;
}

//A função Merge abaixo está incorreta
void Merge(int V[], int p, int q, int r, int U[])
{
        int a = p;
        int b = q+1;

        for(int i = p; i <= q; i++)
                if( b > r || (a <= q && V[a] < V[b]) )
                        U[i] = V[a++];
                else
                        U[i] = V[b++];

        for(int i = p; i <= q; i++)
                V[i] = U[i];
}

void MergeSort(int V[], int primeira, int ultima, int U[])
{
        if(primeira >= ultima) return;

        int p = primeira;
```

```
            int r = ultima;
            int q = (p+r)/2;

            MergeSort(V, p, q, U);
            MergeSort(V, q+1, r, U);
            Merge(V, p, q, r, U);
    }
```

Minutos Restantes: 1960

Usuário: Gabriel Nathan Almeida Silva

Notas:
Q1: 100
Q2: ?
Q3: 100
Total: 67

Entradas:

- Tamanho N do vetor.
- Os N elementos do vetor V.

Saídas:

- Vetor V ordenado.

Exemplo de Entrada:

```
5
7 8 5 3 6
```

Exemplo de saída:

```
3 5 6 7 8
```

Peso: 1

Última tentativa realizada em: 27/11/2019 16:05:07

Tentativas: 2 de 4

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo Nenhum arquivo selecionado Enviar Resposta

Questão 2: Arquivo - BINGO!

Faça um programa que verifique se uma tabela de bingo está premiada. A tabela estará representada por uma **matriz 5x5**, sendo o elemento central representado por **"-1"**, ou seja, posição vazia.

Leia a tabela de um arquivo **"entrada.txt"** juntamente com **10 valores**, que representarão os números sortiados. Se ocorrer o bingo imprima na tela qual em qual posição ocorreu.

- **Imprima: 1 para BINGO na linha;**
- **Imprima: 2 para BINGO na coluna;**
- **Imprima: 3 para BINGO na diagonal principal;**
- **Imprima: 4 para BINGO na diagonal secundária.**
- **Imprima: -1 se não ocorrer BINGO.**

Obs: Não existirá dois casos numa mesma cartela.

Entradas:

- `int tabela[5][5]` - Tabela do BINGO.
- `int sorteio[10]` - Os números sortiados.

Saída:

- Número do caso do BINGO (`int`).

Exemplos de Entradas e Saídas:

Entradas:

```
 1   5  10  15  20
21  16  11   6   2
 7   9  -1  27  32
55   3  54  14  99
43  65  67  13  22

98 20 27 67 43 99 4 6 77 3
```

Saída:

```
4
```

Somente entrada em arquivo!

Peso: 1

Nova Resposta:

Selecione o arquivo com o código fonte do programa que resolve o problema para enviá-lo.

Escolher arquivo Nenhum arquivo selecionado Enviar Resposta

Minutos Restantes: 1960

Usuário: Gabriel Nathan Almeida Silva

Notas:
Q1: 100
Q2: ?
Q3: 100
Total: 67

Questão 3: Arquivo - Xadrez - Rei

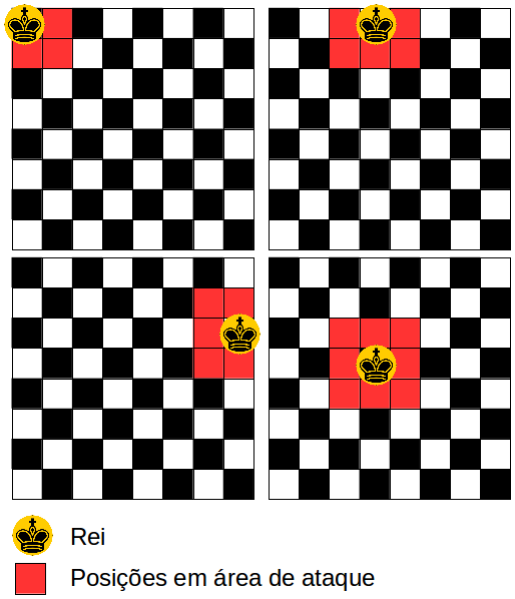
Faça um programa que receba uma matriz 8x8 de um arquivo **"entrada.txt"**. A matriz poderá conter apenas os números **0, 1 ou 2**,

Minutos Restantes: 1960

Usuário: Gabriel Nathan Almeida Silva

Notas:
Q1: 100
Q2: ?
Q3: 100
Total: 67

representando as peças Rei (1), Peão (2) ou local vazio (0) em um tabuleiro de xadrez.



A matriz deve possuir apenas um único número 1, que representa o Rei. Em um jogo de xadrez o Rei pode mover-se em qualquer direção - horizontal, vertical ou diagonal - porém apenas uma casa por vez. O conjunto de posições adjacentes àquela ocupada pelo Rei são denominadas **área de ataque**. Localize o Rei (representado pelo número 1) na matriz lida e verifique quantos Peões (representados pelo número 2) estão em sua área de ataque. Escreva no arquivo **saida.txt** a quantidade de peões que se localizam nessa área de ataque.

Dica: *Podem existir de zero a oito peões posicionados nas redondezas do Rei.*

Dica Adicional: *resolva o problema usando entrada e saída padrão inicialmente, antes de trabalhar com arquivo.*

Entradas:

- `int mat[8][8]` - Tabuleiro do jogo.

Saídas:

- **No arquivo "saida.txt":**: quantos peões se encontram na área de ataque (`int`).

Exemplos de Entradas e Saídas:

Entradas do arquivo "entrada.txt":

```
0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
0 2 0 2 0 0 0 0
0 0 0 2 0 0 2 0
0 0 0 0 2 2 0 2
0 0 0 0 0 1 0 0
0 0 0 0 0 0 0 0
```

Saídas no arquivo "saida.txt":

```
2
```

Entradas do arquivo "entrada.txt":

```
0 0 0 0 0 0 0 0
2 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

```
0 2 0 2 0 0 2 2
0 0 0 2 0 0 2 1
0 0 0 0 0 2 0 2
0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0
```

Saídas no arquivo "saida.txt":

```
4
```

Minutos Restantes: 1960

Usuário: Gabriel Nathan Almeida Silva

Notas:
Q1: 100
Q2: ?
Q3: 100
Total: 67

Peso: 1

Última tentativa realizada em: 28/11/2019 15:16:46

Tentativas: 4 de 4

Nota (0 a 100): 100

Status ou Justificativa de Nota: Nenhum erro encontrado.

Ver Código da Última Tentativa

Você esgotou o máximo de tentativas nesta questão.