BCC202 – Estruturas de Dados I (2020-01)

Departamento de Computação - Universidade Federal de Ouro Preto - MG

TRABALHO PRÁTICO 01

- **Data de entrega: Até 05 de março às 17h..**

- Procedimento para a entrega:.

1. Submissão: via **RunCodes**.
2. Os nomes dos arquivos e os nomes das funções devem ser especificados utilizando boas práticas de programação.
3. Funções auxiliares, complementares aquelas definidas, podem ser especificadas e implementadas, se necessário.
4. A solução deve ser devidamente modularizada e separar a especificação da implementação em arquivos *.h* e *.c* sempre que cabível.
5. Os arquivos a serem entregues, incluindo aquele que contém *main()*, devem ser compactados (*.zip*), sendo o arquivo resultante submetido via **RunCodes**.
6. Caracteres como acento, cedilha e afins não devem ser utilizados para especificar nomes de arquivos ou comentários no código.

- Bom trabalho!

Questão 01 - Revisão BCC201

O jogo da velha é um dos jogos mais antigos da humanidade; os primeiros registros dele são do século I antes de Cristo, no Império Romano. Implemente uma versão simplicada do jogo da velha utilizando uma matriz 3x3 para que o usuário possa jogar contra o computador. A sua meta é fazer do computador o mais esperto possível: a cada jogada do usuário, avalie estrategicamente como o computador irá jogar.

Entrada e Saída

Inicialmente, a entrada é composta pela identificação da célula onde o usuário fará sua primeira jogada: l c, sendo l a idenficação da linha e c da coluna, tal que $0 \leq l, c \leq 2$. Para cada jogada do usuário, será exibida a jogada do computador, também no formato l c. Tão logo termine uma partida, deverá ser exibido o placar: seguido do ganhador: 0 para empate, 1 caso o usuário tenha vencido ou -1 caso o usuário tenha perdido. Ao final das 3 jogadas inciais, a saída será composta pelo placar *numero_partidas_usuario numero_partidas_computador*. O usuário poderá escolher entre parar o jogo (p) ou continuar (c). Caso o usuário escolha continuar, serão executadas mais duas rodadas, ao final das quais será exibido o placar atualizado e assim sucessivamente. Se a escolha for parar, será exibido quem é o vencedor: usuário ou computador.

PONTO EXTRA: Para quem implementar uma interface gráfica para o jogo da velha, fica valendo pontuação extra!

Exemplo de Execução

A seguir, um exemplo de execução do jogo da velha:

Entrada	Saída
0 0 1 1 2 2 1 1 2 2 0 2 1 2 1 1 2 2 0 2 1 2 p	0 1 2 1 0 1 0 0 2 0 0 1 0 0 2 0 3 0

Questão 02 - Tipo Abstrato de Dado (TAD)

Implemente uma versão simplificada do jogo conhecido como *Jokempô* ou *pedra, papel, tesoura*. Neste jogo, o usuário e o computador escolhem entre *pedra, papel* ou *tesoura*. Sabendo que *pedra* ganha de *tesoura*, *papel* ganha de *pedra* e *tesoura* ganha de *papel*, exiba na tela o ganhador das n rodadas: usuário ou computador. Empates também devem ser contabilizados. O número de rodadas, n, pode ser alterado dinamicamente, sendo iniciado com 5 e pode ser acrescido de duas em duas partidas. Ao terminar um conjunto de partidas, o usuário poderá escolher continuar ou parar o jogo. Para essa implementação, assuma que o número 0 representa a *pedra*, 1 representa *papel* e 2 representa *tesoura*. Observação: seu algoritmo deve ser honesto: a escolha feita pelo computador não pode ser realizada baseada na escolha do usuário e vice-versa. Dica: um jogo é composto por jogadores e jogadas, mantendo atualizado o placar. Importante: Cada um dos seus TADs deve conter sua especificação e implementação, conforme visto em sala de aula.

Entrada e Saída

Inicialmente, para cada entrada do usuário (0 para pedra, 1 para papel ou 2 para tesoura), haverá uma saída indicando a jogada do computador (0 para pedra, 1 para papel ou 2 para tesoura) e o resultado da jogada (0 para empate, 1 caso o usuário tenha vencido ou -1 caso o usuário tenha perdido). Ao final das 5 jogadas inciais, a saída será composta pelo placar $numero\_partidas\_usuario$ $numero\_partidas\_computador$. O usuário poderá escolher entre parar o jogo (p) ou continuar (c). Caso o usuário escolha continuar, serão executadas mais duas rodadas, ao final das quais será exibido o placar atualizado e assim sucessivamente. Se a escolha for parar, será exibido quem é o vencedor: usuário ou computador.

Exemplo de Entrada e Saída

A seguir, um possível cenário de execução do Jokempô.

Entrada	Saída
0 1 1 0 1 c 1 2 p	0 0 2 -1 0 1 2 1 1 0 2 1 2 -1 1 1 3 2 Voce ganhou.

Questão 03 - Recursividade e Notação Assintótica

Solicitaram que você construísse um programa simples de criptografia.[1]. Esse programa deve possibilitar o envio de mensagens codificadas. O processo é muito simples. São feitas três passadas em todo o texto, conforme segue:

> Na primeira passada, somente caracteres que sejam letras minúsculas e maiúsculas devem ser deslocadas 3 posições para a direita, segundo a tabela ASCII: letra 'a' deve virar letra 'd', letra 'y' deve virar caractere 'l' e assim sucessivamente. Na segunda passada, a linha deverá ser invertida. Na terceira e última passada, todo e qualquer caractere a partir da metade em diante (truncada) devem ser deslocados uma posição para a esquerda na tabela ASCII. Neste caso, 'b' vira 'a' e 'a' vira ' '.

Por exemplo, se a entrada for "Texto #3", o primeiro processamento sobre esta entrada deverá produzir "Whwr #3". O resultado do segundo processamento inverte os caracteres e produz "3# rw{hW". Por último, com o deslocamento dos caracteres da metade em diante, o resultado final deve ser "3# rvzgV".

Importante: seu programa deve ser composto de funções recursivas para criptografar a mensagem. Para cada função recursiva implementada, esquecifique sua **equação de recorrência** e estime sua complexidade utilizando notação **O** para o pior caso.

Entrada

A entrada contém vários casos de teste. A primeira linha de cada caso de teste contém um inteiro N ($1 \leq N \leq$ 1*104), indicando a quantidade de linhas que o problema deve tratar. As N linhas contém cada uma delas M ($1 \leq M \leq$ 1*103) caracteres.

Saída

Para cada entrada, deve-se apresentar a mensagem criptografada.

Exemplo de Execução

A seguir, um possível cenário de execução do programa para criptografia.

Entrada	Saída
4 Texto #3 abcABC1 vxpdylY .ph vv.xwfxo.fd	3# rvzgV 1FECedc ks. \n{frzx gi.r{hyz-xx

[1]Exercício baseado na atividade disponível em: https://www.urionlinejudge.com.br/judge/pt/problems/view/1024