

Prova 2

Algoritmos e Estruturas de Dados I - turma TW

Professor: Pedro O.S. Vaz de Melo

6 de maio de 2014

Nome: _____

escrevendo o meu nome eu juro que seguirei o código de honra

Código de Honra para este exame:

- Não darei ajuda a outros colegas durante os exames, nem lhes pedirei ajuda;
- não copiarei nem deixarei que um colega copie de mim;
- não usarei no exame elementos de consulta não autorizados.

Informações importantes:

- Em questões que pede um **programa**, este deve ser completo, com bibliotecas (incluindo, quando necessário, a biblioteca **prova2.h**), função **main**, etc. Se deve ser feita uma **função**, somente a função é suficiente. Se deve ser feito um **procedimento**, somente o procedimento é suficiente.
- A interpretação das questões da prova faz parte do critério de avaliação. Caso tenha dúvida sobre a sua interpretação de uma determinada questão, escreva as suas suposições na resolução da mesma.

Referências:

Função/Operador	Descrição	Exemplo
<code>rand()</code>	gera um número aleatório inteiro entre 0 e <code>RAND_MAX</code>	<code>rand()</code> pode gerar 41
<code>RAND_MAX</code>	o maior número possível que pode ser gerado por <code>rand()</code>	<code>RAND_MAX = 32767</code>

1. (10 points) Neste exercício, você deve criar um tipo de dados novo e duas funções para um sistema de gerenciamento de um campeonato de futebol.

a. (1 pt) Defina um novo tipo de dados chamado `Time` com os seguintes campos: `codigo` (`char[4]`), `vitorias` (inteiro), `derrotas` (inteiro) e `empates` (inteiro).

b. (2 pts) Escreva uma função de nome `lerCodigoTime` que recebe um `time` como parâmetro (por referência), lê do teclado o seu código (4 caracteres), e o atribui ao campo `codigo` do `time` que foi passado como parâmetro. Não precisa testar se o usuário digitou 4 caracteres, ou seja, considere que o usuário digitará o código corretamente sempre.

c. (2 pts) Escreva uma função de nome `pontosTime` que recebe um `time` como parâmetro e retorna o número de pontos que o `time` tem. Considere que uma vitória vale 3 pontos e um empate vale 1 ponto.

d. (5 pts) Escreva um procedimento de nome `simulaJogo` que recebe dois `times` como parâmetros (por referência), o primeiro sendo o `mandante`, e o segundo sendo o `visitante`. Desenvolva, a partir da função `rand`, uma maneira de simular um jogo entre essas duas equipes da seguinte maneira: dê vitória ao `mandante` 50% das vezes que a função for chamada; dê vitória ao `visitante` 25% das vezes que a função for chamada; marque empate 25% das vezes que a função for chamada. Atualize os campos `vitorias`, `derrotas` e `empates` das duas equipes de acordo com o resultado da simulação.

2. (8 points) Escreva um programa para simular um campeonato entre 10 equipes e imprimir o código da equipe campeã, ou seja, aquela com o maior número de pontos (considere que haverá apenas uma equipe com a maior pontuação). O campeonato deve ser de pontos corridos e deve haver turno e retorno, ou seja, todas as equipes jogam contra todas as outras duas vezes, uma vez como mandante (em casa) e outra vez como visitante. Dicas: crie um vetor de `times` e use as funções do exercício anterior para preencher os códigos das 10 equipes, simular os jogos entre elas e acessar as suas pontuações.

3. (7 points) Escreva uma função recursiva que receba um valor inteiro x e o imprime invertido. Exemplo: se $x = 123$, a função deve imprimir 321. A função **deve** ter o seguinte protótipo:

```
void imprimeInvertido(int n);
```