Prova 2

Algoritmos e Estruturas de Dados I

Professor: Pedro O.S. Vaz de Melo

17 de novembro de 2015

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.
- As funções implementadas no módulo `prova2.h` podem ser usadas em **qualquer** exercício da prova. Além disso, se você usar uma função do módulo `prova2.h`, considere que ela está implementada de forma correta.

1. (*4 points*) Defina os dois novos tipos de dados descritos abaixo:

a. (*2 pts*) Defina um **novo tipo de dados** para representar uma data. Esse tipo de dados deve ser chamado de `Data` e deve ser capaz de representar uma data a partir do seu ano, mês e dia.

b. (*2 pts*) Defina um **novo tipo de dados** para representar um endereço. Esse tipo de dados deve ser chamado de `Endereco` e deve ser capaz de representar o endereço de uma pessoa localizada em qualquer parte do mundo. Considere que um `Endereco` deve conter todas as informações necessárias para que uma carta seja enviada para o endereço correspondente.

2. (*2 points*) Defina um **novo tipo de dados** para representar uma pessoa. Esse tipo de dados deve ser chamado de `Pessoa` e deve ser capaz de armazenar as seguintes informações sobre uma pessoa: nome, idade, data de nascimento e endereço.

3. (*2 points*) Implemente uma função de nome `preencheData` que recebe uma `Data` como parâmetro **por referência** e preenche seus campos com valores lidos do teclado.

4. (*4 points*) Implemente uma função de nome `retornaIdade` que recebe uma `Data` como parâmetro e retorna quantos anos se passaram desde essa data até o dia de hoje. Nesta questão você deve utilizar a função `Data hoje()`, que retorna uma estrutura `Data` com os seus campos automaticamente preenchidos com valores correspondentes a data atual. Exemplo: se a função `hoje()` retornar `17/11/2015` e se a data passada como parâmetro for `18/11/2013`, a sua função deve retornar 1.

```
int retornaIdade(struct Data nascimento);
```

5. (*4 points*) Implemente uma função de nome `preenchePessoa` que recebe uma `Pessoa` como parâmetro **por referência** e preenche todos os seus campos com valores lidos do teclado. Faça uso das funções dos exercícios anteriores (considere que todas estão implementadas de forma correta). Para o endereço, considere que já existe uma função de protótipo `void preencheEndereco(Endereco *end)` que preenche uma variável de tipo `Endereco` com valores lidos do teclado.

6. (*5 points*) Implemente um procedimento que recebe um vetor de pessoas e o seu tamanho n como parâmetros e **imprime** a cidade da pessoa mais nova. O seu procedimento deve ter o seguinte protótipo:

```
void cidadeMaisNovo(Pessoa p[], int n);
```

7. (*2 points*) Complete o código abaixo, considerando que o programa abaixo deve ler a informação de 50 pessoas do teclado e imprimir a cidade da pessoa mais nova.

```
1:  #include <stdio.h>
2:  #include "prova2.h"
3:
4:  void main(void) {
5:     Pessoa pessoas[50];
6:     int i;
7:     for(i=0; i<50; i++) {
8:        preenchePessoa(_______________);
9:     }
10:    printf("\ncidade da pessoa mais nova:\n");
11:    ____________________________;
12: }
```

8. (*5 points*) Escreva um programa que cria uma matriz de inteiros 100×100 e preenche as suas duas diagonais com o valor 1. O restante da matriz deve ser preenchida com zeros. Abaixo uma matriz 5×5 preenchida desta maneira.

id	n_0	n_1	n_2	n_3	n_4
n_0	1	0	0	0	1
n_1	0	1	0	1	0
n_2	0	0	1	0	0
n_3	0	1	0	1	0
n_4	1	0	0	0	1