P2 Programação II 2012.2 Departamento de Informática/PUC-Rio

Aluno: _____

Matrícula: _____ Turma: _____

	Valor	Nota
Q1	3.0	
Q2	3.0	
Q3	2.5	
Q4	1.5	
Total	10.0	

1. A prova é sem consulta e sem perguntas. A interpretação do enunciado faz parte da prova.
2. A prova deve ser completamente resolvida nas folhas que constam deste caderno, utilizando-se frente e/ou verso;
3. A prova pode ser feita utilizando-se lápis ou caneta (azul ou preta);
4. Dispositivos eletrônicos (celulares, tablets, …) devem ser desligados.
5. Cada questão deve ser resolvida espaço destinado a ela e no verso, quando necessário.

ATENÇÃO: TENHA EM MENTE QUE AS QUATRO QUESTÕES DE SUA PROVA COMPÕEM UM PROGRAMA COM MAIS DE UM MÓDULO, ONDE CADA MÓDULO CORRESPONDE A UM ARQUIVO.

Q1. **(3.0 pts).** Considere que os imóveis de uma imobiliária são cadastrados com 4 informações: código de referência, tipo, ano de construção e bairro. Por exemplo: {101,"apt",1950,"leblon"} e {110,"casa",1950, "botafogo"}. Criando os arquivos necessários (módulos), defina um TAD `Imovel` com 4 campos (`codigo`, `tipo`, `ano`, `bairro`) e 5 funções (*im_cria, im_obtemCodigo, im_obtemTipo, im_obtemAno, im_obtemBairro*), onde: *im_cria* recebe as 4 informações de um imóvel como parâmetros e retorna um ponteiro para `Imovel` cujo espaço na memória foi alocado dinamicamente e cujos campos foram devidamente preenchidos. Não esqueça que ao escrever um TAD é necessário prover uma INTERFACE e uma IMPLEMENTAÇÃO. Crie os módulos (arquivos) que julgar necessário. Indique claramente os nomes dos arquivos (módulos) que você criar, colocando o nome do arquivo como comentário na primeira linha do arquivo. Indique claramente os "includes" que são indispensáveis (**"includes" desnecessários causam perdas de pontos na nota**).

Q1

NAS QUESTÕES 2 E 3, VOCÊ ESTARÁ IMPLEMENTANDO FUNÇÕES COM A FINALIDADE DE MANIPULAR UM VETOR DE IMOVEL. LEMBRE-SE QUE UM MÓDULO DEVE CONTER FUNÇÕES AGREGADAS POR UMA FINALIDADE COMUM. CUIDADO PARA NÃO MISTURAR O TAD DA QUESTÃO 1 COM OUTRO(S) MÓDULO(S). CRIE O(S) MÓDULO(S) QUE JULGAR NECESSÁRIO(S). COLOQUE O NOME DO ARQUIVO COMO COMENTÁRIO NA PRIMEIRA LINHA DO(S) ARQUIVO(S).

Q2 (3.0 pts). Usando um dos métodos de ordenação vistos em sala, o algoritmo BOLHA e o algoritmo ordenação rápida (QUICK SORT), escreva a função **RECURSIVA de ordenação** `vetorImoveis_ordena` que recebe um vetor de ponteiros para TAD `Imovel` e o tamanho deste vetor, e ordena este vetor em ordem ALFABÉTICA de tipo e em ordem DECRESCENTE de ano. Esta função retorna `void`. Você NÃO pode usar a função qsort da biblioteca do C. Indique claramente os "includes" necessários ("includes" a mais ou a menos causam perdas de pontos na nota).

Q3 (2.5 pts). Usando busca binária, escreva a função de busca `vetorImoveis_busca` que recebe um vetor de ponteiros para TAD `Imovel` ordenado conforme a Questão 2, o tamanho deste vetor, um tipo e um ano e retorna um índice do vetor que contenha este tipo e ano. Esta função deve retornar -1 quando não encontrar o tipo e o ano pedidos. Também devem ser indicados claramente os "includes" necessários ("includes" a mais ou a menos causam perdas de pontos na nota).

Q2

Q3

Q4 (1.5 pt). Complete (nas áreas sombreadas) o programa principal a seguir que deve alocar dinamicamente memória para um vetor de ponteiros para `Imovel` e usar as funções de ordenação e busca das Questões 2 e 3. Você deve incluir todas as interfaces que achar necessárias e adequadas (coloque comentários indicando as razões para tal inclusão, como no exemplo do `stdio.h`). Apenas comandos "*includes*" devem constar antes da linha "*int main(void)*". "*includes*" desnecessários causam perdas de pontos na nota. Também declare variáveis necessárias, complete chamadas de função, etc.

```
/* arquivo principal.c */
#include <stdio.h> /* por causa da printf */

int main(void)
{

    int i, n = 5;
    v[0] = im_cria(101,"apt",1990,"leblon");
    v[1] = im_cria(120,"casa",1990,"botafogo");
    v[2] = im_cria(110,"apt",2000,"meier");
    v[3] = im_cria(105,"casa",2000,"meier");
    v[4] = im_cria(130,"apt",2000,"leblon");
    vetorImoveis_ordena(
    printf("Entre tipo e ano: ");
    scanf("%s %d",&tipo,&ano);
    i = vetorImoveis_busca(
    if
        printf("Nao ha' este tipo de imovel\n");
     else
        printf("Bairro: %s\n",
    return 0;
}
```

stdio.h:

```
int scanf (char* formato, ...);
int printf (char* formato, ...);
FILE* fopen (char* nome, char* modo);
int fclose (FILE* fp);
int fscanf (FILE* fp, char* formato, ...);
int fprintf (FILE* fp, char* formato, ...);
char*fgets(char* str, int size, FILE* fp);
int sscanf(char* str, char* formato, ...);
```

stdlib.h:

```
void* malloc (int nbytes);
void free (void* p);
```

math.h:

```
double sqrt (double x);
double pow (double x, double exp);
double cos (double radianos);
double sin (double radianos);
```

string.h:

```
int strlen (char* s);
int strcmp (char* s, char *t);
char* strcpy (char* destino, char* fonte);
char* strcat (char* destino, char* fonte);
char* strdup (char* s);
```