



Aluno: \_\_\_\_\_

Matrícula: \_\_\_\_\_ Turma: \_\_\_\_\_

### Instruções:

- 1) Escreva seu nome completo, matrícula e turma;
- 2) A prova deve ser completamente resolvida nas folhas que constam deste caderno, utilizando-se frente e/ou verso;
- 3) Todas as funções e programas devem ser implementados na linguagem C
- 4) As questões podem ser resolvidas em qualquer ordem;
- 5) As soluções que não forem apresentadas nas páginas a elas destinadas devem ser identificadas com o número da questão a que se referem;
- 6) A prova pode ser feita utilizando-se lápis ou caneta (azul ou preta);
- 7) Todos os dispositivos eletrônicos (celulares, i-pods, etc) devem ser desligados.

### Pontuação:

Questão	Item	Valor	Nota
1	A	1,5	
	B	1,5	
2	-	2,5	
3	-	2,5	
4	-	2,0	
Total		10,0	

## Questão 1

- a) [Valor: 1,5 pontos] Considere uma lista encadeada contendo caracteres, conforme definida pelo tipo a seguir:

```
struct lista {
    char info;
    struct lista* prox;
};

typedef struct lista Lista;
```

Escreva uma função que insere um novo caractere no final da lista, **se a lista já não possuir o caractere**. A função recebe um ponteiro para a lista onde o caractere deve ser inserido e o caractere a ser inserido, e retorna o ponteiro para o início da lista, após a inserção, conforme definido pelo protótipo a seguir:

```
Lista* insereFinal(Lista* l, char c);
```

- b) [Valor: 1,5 pontos] Escreva uma função que recebe duas cadeias de caracteres como parâmetro e retorna uma lista encadeada contendo os caracteres comuns às duas cadeias, sem que haja repetição na lista. O protótipo para a função é dado a seguir.

```
Lista* intersecaoCadeias(char* s1, char* s2);
```

Obs: utilize a função do item (a).

*Solução da Questão 1:*

## Questão 2

[Valor: 2,5 pontos] A estrutura a seguir representa produtos de uma loja virtual

```
struct produto {
    int codigo; /* codigo do produto */
    char nome[101]; /* nome do produto */
    char categoria[51]; /* categoria: cd, dvd, livro, etc. */
    float preco; /* preco do produto */
};

typedef struct produto Produto;
```

Escreva uma função que recebe um vetor de ponteiros para os produtos de uma loja (*vet*), o número de elementos no vetor (*n*), o nome de uma categoria (*categoria*) e um preço (*preco*), e retorna um vetor, alocado dinamicamente, contendo ponteiros para os produtos da categoria especificada com preço maior ou igual ao passado como parâmetro. A função recebe um quinto parâmetro (*tam*), do tipo ponteiro para inteiro, que deve conter o número de elementos do vetor retornado.

```
Produto** selecionaProdutos(Produto** vet, int n, char* categoria,
                           float preco, int* tam);
```

Observações:

- 1) O vetor retornado deve ser alocado com o tamanho exato necessário.
- 2) Não existe valor NULL nos *n* primeiros elementos do vetor *vet*.

*Solução da Questão 2:*

### Questão 3

[Valor: 2,5 pontos] Uma estrutura C é utilizada para descrever os assinantes de uma revista:

```
struct data {
    int dia;
    int mes;
    int ano;
};

typedef struct data Data;

struct assinante {
    char cpf[12]; /* cpf do assinante */
    char nome[151]; /* nome do assinante */
    Data dataAssinatura; /* data de inicio da assinatura */
};

typedef struct assinante Assinante;
```

Um sistema mantém os assinantes da revista ordenados cronologicamente pela data do início da assinatura. Se dois ou mais assinantes possuem a mesma data de assinatura, os assinantes são organizados aleatoriamente.

**Utilizando o algoritmo de busca binária**, escreva uma função que imprime na tela todos os clientes que fizeram uma assinatura da revista em uma determinada data, conforme o protótipo a seguir.

```
void imprimeAssinantes(Assinante** vet, int n, Data d);
```

A função recebe um vetor de ponteiros para a estrutura assinante (*vet*), o número de ponteiros para assinantes no vetor (*n*) e a data a ser pesquisada (*d*). Sua função deve imprimir a data pesquisada e, em seguida, o cpf e o nome de cada assinante, conforme exemplificado a seguir:

```
Assinantes de 10/06/2010
33445566778 Ricardo Silva
11122233345 Maria das Dores
99988877700 Luiza Chaves
```

Se não houver assinantes para a data especificada, a função deve imprimir a mensagem “Nenhum assinante encontrado.”.

```
Assinantes de 11/06/2010
Nenhum assinante encontrado.
```

Considere que os *n* primeiros elementos do vetor *vet* são diferentes de NULL.

*Solução da Questão 3:*

*Não separe as folhas deste caderno.*

## Questão 4

[Valor: 2,0 pontos] Considere uma *árvore binária de busca* que armazena cadeias de caracteres, ordenada de tal forma que os valores dos nós da subárvore à esquerda são alfabeticamente anteriores ao valor da raiz, e os valores dos nós da subárvore à direita são alfabeticamente posteriores ou iguais ao valor da raiz. O tipo que representa um nó da árvore é dado por:

```
struct arv {
    char* palavra;
    struct arv *esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Escreva uma função que recebe como parâmetros o ponteiro *a* (para a raiz da *árvore binária de busca* descrita) e uma palavra *p* e, tirando proveito da ordenação da árvore, retorna o número de palavras alfabeticamente posteriores à palavra procurada. A função deve obedecer ao seguinte protótipo:

```
int contaPalavrasMajores(Arv* a, char* p);
```



*Solução da Questão 4:*

*Não separe as folhas deste caderno.*

## Protótipos de funções que podem ser úteis:

### **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, ...);
```

### **math.h:**

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

### **stdlib.h:**

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