

INF 1620 – P4 - 06/07/04	Questão 1
Nome:	
Matrícula:	Turma

- (a) Escreva uma função, que dado um número inteiro n , verifique se é ou não primo. A função deve retornar 1 se o número dado for primo e 0 se não for. O protótipo da função é dado por:

```
int primo (int n);
```

- (b) Escreva um programa completo que capture dois valores inteiros u e v , fornecidos pelo usuário via teclado, e imprima quantos números primos existem entre u e v , inclusive. Este programa deve fazer uso da função do item anterior.

INF 1620 – P4 - 06/07/04	Questão 2
Nome:	
Matrícula:	Turma

Escreva uma função para retirar o sufixo de uma cadeia de caracteres. A função deve receber como parâmetro de entrada uma cadeia de caracteres e o comprimento do sufixo que deve ser retirado. Esta função deve retornar uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo a cadeia original sem o sufixo. Por exemplo, se for passada a cadeia “Estruturas” e o número 7, a função deve retornar uma nova cadeia “Est”. A assinatura da função deve ser:

```
char* retira_sufixo (char* s, int n);
```

Se o comprimento do sufixo for maior do que o comprimento da cadeia original, a função deve retornar NULL.

INF 1620 – P4 - 06/07/04	Questão 3
Nome:	
Matrícula:	Turma

Considere a implementação de uma lista encadeada para armazenar um cadastro de alunos. O tipo que representa a lista é dado a seguir:

```
struct lista {
    int mat;
    char nome[81];
    char email[41];
    struct lista *prox;
};

typedef struct lista Lista;
```

Escreva uma função que, dados o ponteiro para o primeiro elemento da lista e um número de matrícula, retire o aluno correspondente da lista. A função deve retornar 1 se o aluno for retirado com sucesso e retornar 0 se o número de matrícula fornecido não for encontrado na lista. A função deve obedecer ao protótipo:

```
int retira (Lista* l, int mat);
```

INF 1620 – P4 - 06/07/04	Questão 4
Nome:	
Matrícula:	Turma

Considere uma *árvore binária de busca* que armazena valores inteiros, onde os valores associados aos nós da sub-árvore à esquerda são menores que o valor associado à raiz e que os valores dos nós da sub-árvore à direita são maiores. O tipo que representa um nó da árvore é dado por:

```
struct arv {
    int val;
    struct arv* esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Escreva uma função que retorne o número de nós da árvore cuja informação seja maior que um dado valor x . A função deve tirar proveito da ordenação da árvore e obedecer ao seguinte protótipo:

```
int maiores_x (Arv* a, int x);
```

INF 1620 – P4 - 06/07/04	Questão 5
Nome:	
Matrícula:	Turma

Considere um arquivo texto que descreve um conjunto de retângulos, triângulos e círculos. Cada linha do arquivo contém a descrição de uma figura. O primeiro caractere da linha indica seu tipo: *r* para retângulo, *t* para triângulo e *c* para círculo. Esse caractere é seguido pelos valores, números reais, da base e da altura para retângulos e triângulos e do raio para círculos. Considere ainda que não existem linhas em branco no arquivo. Um exemplo desse formato é mostrado abaixo.

r	2.0	4.3
t	4.0	5.0
c	3.0	
t	6.0	0.5
r	1.0	1.02

Escreva um programa completo que leia o arquivo “entrada.txt”, que contém as descrições das figuras no formato descrito acima, e imprima na tela o valor do somatório das áreas das figuras listadas no arquivo. Se não for possível abrir o arquivo, o programa deve ter como saída a mensagem “ERRO”. Se não existir nenhuma figura no arquivo (arquivo existente, mas vazio), deve-se imprimir o valor zero.

RASCUNHO

Respostas nesta folha não serão consideradas.

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);
```

string.h:

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

stdlib.h:

```
void* malloc (int nbytes);
void free (void* p);
void qsort (void *vet, int n, int tam, int (*comp) (const void*, const void*));
```

Não separe as folhas deste caderno.