

INF 1620 – P4 - 11/12/04	Questão 1
Nome:	
Matrícula:	Turma

- (a) Escreva uma função que calcule o volume e a área da superfície de um cone, dados seu raio (r) e sua altura (h). O protótipo da função deve ser:

```
void calc (float r, float h, float* area, float* volume);
```

Fórmulas:

$$volume = \frac{1}{3}pr^2h$$

$$area = prs$$

$$s = \sqrt{r^2 + h^2}$$

$$p = 3.14159265$$

- (b) Escreva um programa completo que capture dois valores de ponto flutuante, fornecidos pelo usuário via teclado, correspondentes ao raio e altura de um cone, e imprima o volume e a área da superfície desse cone. Este programa deve fazer uso da função do item anterior.

INF 1620 – P4 - 11/12/04	Questão 2
Nome:	
Matrícula:	Turma

Escreva uma função para copiar os n primeiros caracteres de uma cadeia de caracteres. A função deve receber como parâmetros de entrada uma cadeia de caracteres e o número de caracteres que se deseja copiar (n), e deve retornar uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo apenas os n primeiros caracteres da cadeia original. Por exemplo, se for passada a cadeia “Estruturas” e o número 3, a função deve retornar uma nova cadeia “Est”. A assinatura da função deve ser:

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

Se o parâmetro n for maior ou igual ao comprimento da cadeia original, a função deve retornar uma cópia completa da cadeia original. O espaço de memória alocado para a nova cadeia deve ser do tamanho exatamente necessário para armazená-la.

INF 1620 – P4 - 11/12/04	Questão 3
Nome:	
Matrícula:	Turma

Considere a implementação de uma lista encadeada para armazenar as notas dos alunos de uma turma. O tipo que representa a lista é dado a seguir:

```
struct lista {
    char nome[81];
    float nota;
    struct lista *prox;
};

typedef struct lista Lista;
```

Pede-se:

- (a) Assumindo que a lista está em ordem alfabética pelo nome do aluno, escreva uma função para inserir um novo elemento na lista que mantenha essa ordenação. O protótipo da função deve ser:

```
Lista* insere (Lista* l, char* nome, float nota);
```

- (b) Escreva uma função para retornar a quantidade de alunos aprovados na turma, assumindo que a média para passar é 5. Se a lista for vazia, deve-se retornar o valor zero. O protótipo da função deve ser:

```
int aprovados (Lista* l);
```

INF 1620 – P4 - 11/12/04	Questão 4
Nome:	
Matrícula:	Turma

Considere a implementação de um cadastro de produtos usando uma árvore binária de busca. A estrutura é ordenada segundo o identificador numérico dos produtos, isto é, os nós à esquerda da raiz representam produtos com identificadores menores que o da raiz, e os nós à direita produtos com identificadores maiores. Os elementos da árvore são definidos por:

```
struct arv {
    int id;           /* identificador */
    int quant;       /* quantidade do produto */
    struct arv* esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Pede-se

- (a) Tirando proveito da ordenação da estrutura, escreva uma função que, dado o ponteiro para o nó raiz e um código numérico, tenha como valor de retorno a quantidade do produto dado. Se o produto não estiver presente, a função deve ter zero como valor de retorno. O protótipo dessa função deve ser:

```
int quantidade (Arv* a, int id);
```

- (b) Escreva uma função para imprimir na tela os identificadores, em ordem *decrecente*, de todos os produtos armazenados na árvore. O protótipo dessa função é dado por:

```
void imprime (Arv* a);
```

INF 1620 – P4 - 11/12/04	Questão 5
Nome:	
Matrícula:	Turma

Considere um arquivo que contenha as coordenadas, x e y , de pontos no plano, com cada ponto ocupando uma linha do arquivo. Um exemplo desse arquivo é dado a seguir:

1.2	3.2
2.4	1.5
3.1	2.3
9.1	5.5
2.1	4.9
7.5	4.8

Escreva uma função que leia o conteúdo de um arquivo no formato descrito acima e salve em um outro arquivo no mesmo formato apenas os pontos do arquivo de entrada que se encontram dentro de uma determinada região retangular. Sua função deve receber como parâmetros o nome do arquivo de entrada, o nome do arquivo de saída, e as coordenadas de dois pontos que determinam a região do plano desejada, de acordo com o protótipo a seguir:

```
int filtra_pontos (char* entrada, char* saida, float x1, float x2,  
                 float y1, float y2);
```

O arquivo de saída deve conter apenas os pontos cujas as coordenadas x e y obedecem a regra: $x1 < x < x2$ e $y1 < y < y2$. Por exemplo, supondo que o arquivo de entrada fornecido tenha o conteúdo do exemplo acima e os valores dos parâmetros $x1$, $x2$, $y1$ e $y2$ sejam respectivamente 3.0, 9.5, 2.0 e 5.0, o arquivo de saída gerado pela sua função teria o seguinte conteúdo:

3.1	2.3
7.5	4.8

Sua função deve retornar 0 se ocorrer erro na abertura dos arquivos, e 1 caso contrário. Arquivos de entrada vazios devem gerar arquivos de saída vazios.

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 separeas folhas deste caderno.