

INF 1620 – P4 - 09/07/03	Questão 1
Nome:	
Matrícula:	Turma

Um ponto geométrico no espaço 2D é representado pelas coordenadas reais x e y . Considere uma aplicação que precisa manipular pontos, através da implementação de duas funções. A primeira recebe valores x e y e cria dinamicamente um tipo ponto correspondente. A segunda recebe dois pontos como parâmetros e retorna a distância entre eles. As assinaturas dessas funções são apresentadas abaixo:

```
/* definição de tipo */
typedef struct ponto Ponto;

/* funções */
Ponto* cria (double x, double y);
double distancia (Ponto* a, Ponto* b);
```

Escreva um código C que defina a estrutura `struct ponto` e faça a implementação das duas funções.

Formulário:

Fórmula da distância d entre dois pontos a e b : $d = \sqrt{(a_x - b_x)^2 + (a_y - b_y)^2}$

Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P4 - 09/07/03	Questão 2
Nome:	
Matrícula:	Turma

Escreva uma função em C para inverter uma cadeia de caracteres. A função deve receber como parâmetro de entrada uma cadeia de caracteres e retornar uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo a cadeia original invertida, de trás para frente. Por exemplo, se for passada a cadeia “Puc-Rio”, a função deve retornar uma nova cadeia “oiR-cuP”. A assinatura da função deve ser:

```
char* inverte (char* s);
```

INF 1620 – P4 - 09/07/03	Questão 3
Nome:	
Matrícula:	Turma

Considere uma lista encadeada que represente um conjunto de alunos matriculados em disciplinas de um determinado curso. Cada elemento da lista contém um código alfa-numérico da disciplina, o nome do aluno e a nota obtida pelo aluno na disciplina. O tipo que representa o nó da lista é dado por:

```
struct lista {
    char cod[8];          /* código da disciplina */
    char nome[81];       /* nome do aluno */
    float nota;          /* nota do aluno na disciplina */
    struct lista* prox;  /* ponteiro para próximo elemento */
};
typedef struct lista Lista;
```

Escreva uma função que, dado o código de uma disciplina, tenha como valor de retorno a média das notas obtidas pelos alunos nessa disciplina. Se na lista não existir alunos na disciplina dada, a função deve ter zero como valor de retorno. O protótipo dessa função é dado por:

```
float media (Lista* l, char* cod);
```

INF 1620 – P4 - 09/07/03	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 - 09/07/03	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. Considere ainda que pontos iguais aparecem em posições consecutivas nesse arquivo. Um exemplo desse arquivo é dado a seguir:

1.2	3.2
2.4	1.5
2.4	1.5
2.4	1.5
3.1	2.3
3.1	4.5
3.1	4.5
7.5	4.8

Escreva um programa completo que leia o conteúdo do arquivo “entrada.txt” no formato descrito acima e salve um arquivo “saida.txt” no mesmo formato, mas eliminando as repetições de pontos iguais. Por exemplo, se aplicado ao arquivo mostrado acima, o programa geraria o seguinte arquivo de saída:

1.2	3.2
2.4	1.5
3.1	2.3
3.1	4.5
7.5	4.8

Se ocorrer erro na abertura dos arquivos, deve-se imprimir a mensagem “ERRO” na tela.

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, ...);
```

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);
```

Não separe as folhas deste caderno.