INF 1620 – P3 - 29/06/04	Questão 1
Nome:	
Matrícula:	Turma

Considere um arquivo texto que descreve um conjunto de retângulos e círculos. Cada linha do arquivo contém a descrição de uma figura. O primeiro caractere da linha indica seu tipo: `r` para retângulo e `c` para círculo. Esse caractere é seguido pelos valores, números reais, da base e da altura para retângulos 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  4.0
c  2.0
r  1.0  1.5
r  6.0  0.5
c  1.0
```

Escreva um programa completo que leia o arquivo "entrada.txt", que contém as descrições das figuras no formato descrito acima, e crie um arquivo texto com o nome "saida.txt" que tenha a mesma lista de figuras, *na mesma ordem de entrada*, mostrando em cada linha apenas duas informações: o caractere que identifica a figura (`r` ou `c`) e a área da figura. O número de casas decimais usado para imprimir os valores das áreas pode ser qualquer. Se o arquivo acima representar a entrada, a saída deveria conter:

```
r  8.00
c  12.56
r  1.50
r  3.00
c  3.14
```

Se o arquivo de entrada não existir, o programa deve imprimir "ERRO" na tela. Considere que sempre será possível criar o arquivo de saída.

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 – P3 - 29/06/04	Questão 2
Nome:	
Matrícula:	Turma

Considere o tipo que representa uma lista de figuras geométricas (retângulos, triângulos e círculos). O tipo para representar cada uma das figuras é mostrado a seguir:

```
struct figura {
   char tipo;      /* tipo: c = circ, t = triang, r = ret */
   float v1;       /* valor do raio (c) ou da base (t ou r) */
   float v2;       /* valor da altura (t ou r) */
   float area;     /* área de figura (ja'calculada) */
};
typedef struct figura Figura;
```

Utilizando a função `qsort` da biblioteca padrão, escreva uma função para ordenar um vetor de figuras, em ordem crescente de tipo, usando como critério de desempate a ordem crescente de área. Desta forma, o vetor resultante terá como primeiros elementos os círculos em ordem crescente de área, seguidos dos retângulos e, por fim, seguidos dos triângulos. A função recebe como parâmetros o número de figuras e um vetor de figuras. O protótipo da função deve ser:

```
void ordena (int n, Figura* v);
```

Obs: a função `ordena` consistirá apenas da chamada da função `qsort`, fazendo uso de uma função auxiliar para comparar os elementos do vetor, que deve ser escrita.

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 – P3 - 29/06/04	Questão 3
Nome:	
Matrícula:	Turma

Considere o tipo que representa um produto de um estoque dado por:

```
struct produto {
   int id;          /* código de identificação */
   char nome[81];   /* nome do produto */
   int quant;       /* quantidade disponível no estoque */
   float preco;     /* preço de venda */
};
typedef struct produto Produto;
```

Considere um vetor que armazena ponteiros para os produtos presentes em um estoque ordenado (ordem crescente) pelo código de identificação de cada produto. Sem usar funções da biblioteca padrão, escreva uma função que faça um a *busca binária* no vetor. Essa função recebe como parâmetros de entrada o número de produtos, o vetor e o código do produto que se deseja buscar. A função deve ter como valor de retorno o ponteiro para o produto cujo código é igual ao código fornecido. Se não existir um produto com o código fornecido, a função deve ter `NULL` como valor de retorno. O protótipo da função deve ser dado por:

```
Produto* busca (int n, Produto** vet, int id);
```

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 – P3 - 29/06/04	Questão 4
Nome:	
Matrícula:	Turma

Considere uma estrutura de árvore binária de busca para armazenar dados relativos a alunos definida pelo tipo abaixo:

```
struct arv {
   int mat;
   char nome[81];
   char email[41];
   struct arv* esq;
   struct arv* dir;
};
typedef struct arv Arv;
```

Considerando que esta árvore usa o *nome do aluno como chave de busca*, implemente uma função que insira um novo aluno na árvore. Considere que o nome do aluno a ser inserido ainda não existe armazenado na árvore, evitando ocorrências duplicadas. A função deve obedecer ao protótipo, retornando a raiz da árvore após a inserção:

```
Arv* insere (Arv* a, int mat, char* nome, char* email);
```

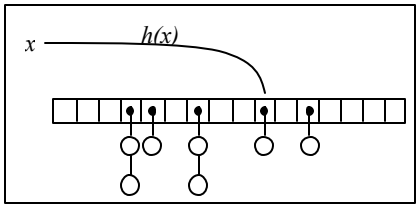
Não separe as folhas deste caderno. Todas as folhas devem ter seu nome. Responda cada questão na folha correspondente. Use o verso se necessário.

INF 1620 – P3 - 29/06/04	Questão 5
Nome:	
Matrícula:	Turma

Considere um cadastro de alunos armazenado numa tabela de dispersão (*hash*) que usa como chave de busca o número de matrícula. Considere também a existência de uma função de *hash* que, dado o número de matrícula, retorna o índice na tabela. O protótipo dessa função é dado por:

```
int hash (int mat);
```

O tratamento de colisão é feito utilizando-se listas encadeadas para armazenar os elementos que colidem. Desta forma, cada posição na tabela representa um ponteiro para o primeiro nó de uma lista encadeada, conforme ilustra a figura ao lado.



Considerando as seguintes declarações relativas à tabela *hash* dos alunos:

```
struct aluno {
   char nome[81];
   int matricula;
   float cr;
   struct aluno* prox;
};
typedef struct aluno Aluno;

#define N 127
typedef Aluno* Hash[N];
```

Implemente uma função que retire um aluno da tabela *hash*. A função deve receber como parâmetros a tabela hash e o número de matrícula do aluno a ser retirado. Se o aluno especificado for retirado com sucesso, a função deve retornar o valor 1. Se o aluno não existir na tabela, a função deve retornar 0. O protótipo é dado por:

```
int retira (Hash tab, int mat);
```

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

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.