INF 1620 – P2 - 17/05/08	Questão 1
Nome:	
Matrícula:	Turma

Considere um cadastro de produtos de um estoque, com as seguintes informações:

- Código de Identificação do produto: representado por um número inteiro
- Nome do produto: com até 50 caracteres
- Preço de venda: representado por um número real
- Estoque mínimo: representado por um número inteiro
- Quantidade disponível no estoque: representado por um valor inteiro

a) Defina uma estrutura em C, denominada **produto**, que tenha os campos apropriados para guardar as informações de um produto, conforme descrito acima.

b) Escreva uma função que receba um vetor de estruturas **produto** (`Produtos`)**,** o número de produtos cadastrados (`N_Produtos`), e retorne um novo vetor de estruturas **produto** alocado dinamicamente contendo os produtos que precisam ser reabastecidos e a quantidade de produtos a serem reabastecidos (`N_Prod_Compras`).

A função deve ter o seguinte protótipo:

```
struct produto * Prod_Compras(struct produto * Produtos,
                              int N_Produtos, int * N_Prod_Compras);
```

Observação:

1- Produtos que precisam ser reabastecidos são aqueles cujo estoque atual é menor que o mínimo.

2- A quantidade de produtos a serem reabastecidos expressa o comprimento do vetor retornado.

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 – P2 - 17/05/08	Questão 2
Nome:	
Matrícula:	Turma

Uma transportadora utiliza uma matriz para representar os custos de transporte entre cidades. Escreva uma função que dado um vetor com as cidades na ordem que deverão ser visitadas, o número de cidades visitadas no percurso, a matriz de custos e a dimensão da matriz, retorne o custo total do itinerário.

A função deve ter o seguinte protótipo:

```
int Custo_Cidades (int* Cidades,int N_Cidades, int* M, int n);
```

Dica: Vale a pena observar que a matriz M, representada como um vetor simples, sempre será quadrada (n x n); e o custo de transporte entre as cidades i e j é dado pelo elemento Mij da matriz.

Exemplo: Considerando a seguinte matriz de custos

$$M_{4x4} = \begin{pmatrix} 1 & 15 & 20 & 33 \\ 75 & 1 & 10 & 400 \\ 23 & 12 & 1 & 8 \\ 17 & 31 & 25 & 1 \end{pmatrix}$$

O custo do itinerário {0, 3, 1, 2, 1, 0} é calculado da seguinte forma:

$M_{03} + M_{31} + M_{12} + M_{21} + M_{10} = 33 + 31 + 10 + 12 + 75 = 161$

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 – P2 - 17/05/08	Questão 3
Nome:	
Matrícula:	Turma

Para organizar seu acervo de filmes, uma locadora armazena o código de controle, o título, ano da filmagem, o nome do diretor e o país de origem do filme. Esse fichamento é representado por um vetor de ponteiros para a estrutura abaixo.

```
struct filme{
        int Codigo;
        char Titulo[61];
        char Direcao[81];
        char Origem[6];
        int Ano;
};

typedef struct filme Filme;
```

Usando uma das técnicas de ordenação vistas no curso, escreva uma função para colocar uma lista de filmes em ordem decrescente do ano da filmagem e, em caso de empate, ordenar pela ordem alfabética dos títulos dos filmes. A função deve receber como parâmetros o vetor que armazena ponteiros para estruturas do tipo **Filme** e o número de filmes, de acordo com o protótipo definido a seguir:

```
void Ordena_Filmes(Filme** Acervo, int n);
```

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 – P2 - 17/05/08	Questão 4
Nome:	
Matrícula:	Turma

Considerando o tipo **Modalidade**, descrito abaixo, e que o vetor que armazena ponteiros para estruturas do tipo **Modalidade** está ordenado em ordem decrescente pelo código da Modalidade esportiva, escreva uma função que faça a busca binária neste vetor.

```
struct modalidade
{
        int Codigo;
        char Nome_Modalidade[81];
        int Medalhas_Ouro;
        int Medalhas_Prata;
        int Medalhas_Bronze;
};
typedef struct modalidade Modalidade;
```

Essa função deve receber como parâmetro o vetor, o número de modalidades e o código da modalidade a ser encontrada, e deve ter como valor de retorno um ponteiro para o registro da modalidade procurada. Se não existir a Modalidade esportiva com código fornecido, a função deve retornar **NULL**. Sua função **não** deve usar a função bsearch da biblioteca padrão da linguagem C, e deve ter o seguinte protótipo:

```
Modalidade* Busca_Modalidade(Modalidade** VetorMod, int n, int Codigo);
```

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);
double cos (double radianos);
double sin (double radianos);
```

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.