

INF 1620 – P2 - 19/05/07	Questão 1
Nome:	
Matrícula:	Turma

Considere a existência de dois arquivos textos. No primeiro arquivo, denominado *nomes.txt*, encontra-se uma lista de nomes de alunos, um por linha (cada nome tem, no máximo, 80 caracteres). No segundo arquivo, denominado *notas.txt*, encontra-se a lista das notas dos alunos. Para cada linha do primeiro arquivo, existe uma linha no segundo arquivo com as três notas obtidas pelo aluno em uma disciplina. Não existem linhas em branco nos arquivos, e o número de linhas existentes nos dois arquivos é igual. Um exemplo destes arquivos é mostrado a seguir:

<i>nomes.txt</i>	<i>notas.txt</i>
Fulano Pereira	9.1 7.5 6.8
Beltrano Silva	5.6 4.6 7.0
Sicrano Santos	7.5 4.6 5.8
Fulana Souza	8.0 8.0 6.0

Escreva um programa completo que leia os conteúdos de dois arquivos (“nomes.txt” e “notas.txt”) com o formato acima e crie um terceiro arquivo, com o nome “saida.txt”, com a lista dos alunos com suas respectivas médias obtidas nas provas: cada linha do arquivo de saída deve conter o nome do aluno seguido da média correspondente, com duas casas decimais. Se os dois arquivos acima fossem usados num exemplo, a saída obtida seria:

<i>saida.txt</i>
Fulano Pereira 7.80
Beltrano Silva 5.73
Sicrano Santos 5.97
Fulana Souza 7.33

Pode-se considerar que sempre será possível abrir os arquivos.

INF 1620 – P2 - 19/05/07	Questão 2
Nome:	
Matrícula:	Turma

a) Reproduza na folha da prova como seriam as transformações do vetor de inteiros
7 10 5 8 1 3
caso ele fosse submetido a um algoritmo bolha de ordenação crescente, até que as 3 últimas posições do vetor já estejam com seus valores definitivos. Ou seja, não precisa fazer a ordenação até que todo o vetor esteja ordenado.

b) Utilizando o algoritmo de ordenação **bolha**, implemente uma função que compute os 3 maiores valores de um vetor cujo protótipo é:

```
void maiores(int n, int* v, int* maior, int* segundo_maior,  
            int* terceiro_maior);
```

Obs.: Considere que o vetor sempre terá pelo menos 3 elementos.

INF 1620 – P2 - 19/05/07	Questão 3
Nome:	
Matrícula:	Turma

Considere um cadastro de pessoas que armazena nome e data de nascimento (representada por três inteiros: dia, mês, ano). O cadastro é representado por um vetor de ponteiros para a estrutura abaixo, ordenado em ordem crescente de data de nascimento:

```
struct pessoa {
    char nome[81];          /* nome */
    int dia, mes, ano;     /* data de nascimento */
};
typedef struct pessoa Pessoa;
```

Aplicando a técnica de busca binária, implemente uma função que verifique se alguém nasceu numa determinada data. A função deve receber o número de elementos no vetor, o vetor de ponteiros e a data que se deseja fazer a busca. Caso seja encontrada uma pessoa que tenha nascido na data especificada, a função deve retornar o ponteiro para a estrutura `Pessoa` correspondente. Se existir mais de uma pessoa que tenha nascido na data procurada, a função deve retornar o ponteiro para a primeira estrutura encontrada. Se não existir uma pessoa no cadastro que tenha nascido na data especificada, a função deve retornar `NULL`. A função deve obedecer ao seguinte protótipo:

```
Pessoa* busca (int n, Pessoa** v, int dia, int mes, int ano);
```

INF 1620 – P2 - 19/05/07	Questão 4
Nome:	
Matrícula:	Turma

Seja uma dada matriz quadrada A de tamanho n . A matriz complementar referente ao elemento $a_{i,j}$ é a matriz quadrada de tamanho $n-1$, que resulta de suprimir a linha i e a coluna j em A . Por exemplo, considerando a matriz a seguir,

$$\begin{bmatrix} 3 & 0 & 1 \\ 12 & 4 & 11 \\ 1 & 17 & 5 \end{bmatrix}$$

a matriz complementar referente ao elemento $a_{1,0}$ é a matriz:

$$\begin{bmatrix} 0 & 1 \\ 17 & 5 \end{bmatrix}$$

Escreva uma função que dada uma certa matriz e os índices de um elemento, cria a matriz complementar a esse elemento. A função recebe como parâmetros um ponteiro para a matriz original, representada por um vetor simples, o tamanho n da matriz original e os índices i e j do elemento de referência e deve retornar um ponteiro para a nova matriz obtida, também representada por um vetor simples e que deve ser alocado dinamicamente. A matriz original não pode ser modificada. Essa função deve ter o seguinte protótipo:

```
int* complementar(int n, int i, int j, int* mat);
```

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 fgetc (FILE* fp);
int fputc (int c, 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*));
void* bsearch (void* info, void *vet, int n, int tam,
              int (*comp) (const void*, const void*));
```

Não separe as folhas deste caderno.