

INF 1620 – P2 - 28/10/06	Questão 1
Nome:	
Matrícula:	Turma

Considere um arquivo texto com as notas dos alunos de uma disciplina. Cada linha do arquivo contém a matrícula de um aluno (cadeia de nove caracteres), seguida pelos valores de suas três notas (P1, P2 e P3). O arquivo pode conter eventuais linhas em branco. Um exemplo desse formato é mostrado a seguir.

```
0510087-2 2.0 4.3 6.5
0420324-3 7.0 8.2 8.6
0510478-5 6.0 7.5 7.8
0510256-8 3.0 0.5 4.2
```

Escreva um programa completo que leia o arquivo “entrada.txt”, que contém as notas dos alunos de uma disciplina no formato descrito acima, e escreva no arquivo texto “saida.txt” o número de matrícula de cada aluno aprovado na disciplina com sua respectiva média. Considere que um aluno está aprovado se tiver média maior ou igual a 5.0, e que a média é calculada de acordo com a fórmula $(P1+P2+P3)/3$. Para o exemplo acima, o seu programa deveria criar um arquivo “saida.txt” com o seguinte conteúdo:

```
0420324-3 7.9
0510478-5 7.1
```

Observe que as médias são escritas no arquivo “saida.txt” com apenas uma casa decimal. Se não for possível abrir um dos arquivos, o programa deve imprimir a mensagem “ERRO” e terminar sua execução. Para um arquivo de entrada vazio, o programa deve gerar um arquivo de saída vazio.

INF 1620 – P2 - 28/10/06	Questão 2
Nome:	
Matrícula:	Turma

Considere um tipo que representa um aluno de uma disciplina, definido pela estrutura a seguir:

```
struct aluno {
    char mat[10];          /* matrícula do aluno */
    char nome[81];        /* nome do aluno */
    float p1;             /* nota da P1 */
    float p2;             /* nota da P2 */
    float p3;             /* nota da P3 */
};
typedef struct aluno Aluno;
```

Escreva uma função que imprima em ordem decrescente de média o nome e a média dos alunos que foram **aprovados** na disciplina. Considere que um aluno está aprovado se tiver média maior ou igual a 5.0, e que a média é calculada de acordo com a fórmula $(P1+P2+P3)/3$. Caso dois alunos possuam médias iguais, seus dados devem ser impressos seguindo a ordem alfabética de seus nomes. A função deve receber como parâmetros o número total de alunos e um vetor de estruturas do tipo `Aluno`, onde estão armazenados os dados dos alunos, de acordo com o protótipo definido a seguir:

```
void imprime_aprovados (int n, Aluno* vet);
```

Dica: sua função pode primeiro ordenar o vetor em ordem decrescente de médias (tendo como critério de desempate o nome do aluno), usando alguma das técnicas de ordenação vistas no curso.

INF 1620 – P2 - 28/10/06	Questão 3
Nome:	
Matrícula:	Turma

Considere um vetor de inteiros cujos valores estão armazenados em ordem crescente. Usando como base o algoritmo de busca binária, escreva uma função que, dado o vetor e um valor inteiro x , retorne o elemento do vetor que possui valor mais próximo de x . Caso x seja equidistante de dois elementos do vetor, sua função deve retornar o valor do menor deles. Por exemplo, considerando o vetor $\{3,7,10,14,16\}$, sua função deveria retornar:

- 10 para o caso de $x = 11$
- 3 para o caso de $x = 5$
- 14 para o caso de $x = 14$
- 14 para o caso de $x = 13$

Considere que o valor de x sempre é maior ou igual ao primeiro elemento do vetor e menor ou igual ao último elemento do vetor. Sua função deve ter o seguinte protótipo:

```
int mais_proximo (int n, int* vet, int x);
```

INF 1620 – P2 - 28/10/06	Questão 4
Nome:	
Matrícula:	Turma

Considere uma matriz $m \times n$ que armazena apenas números inteiros não-negativos. Escreva uma função que receba como parâmetros a matriz (representada por um vetor simples) e suas dimensões m e n , e retorne o somatório dos elementos da linha cujo valor é o maior. Por exemplo, para a matriz

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

sua função deve retornar 27. Sua função deve obedecer ao seguinte protótipo:

```
int maior_somatorio (int m, int n, int* matriz);
```

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.