

INF 1620 – P4 – 11/12/06	Questão 1
Nome:	
Matrícula:	Turma

Considere que o cálculo da multa para o pagamento de um determinado imposto varia de acordo com a tabela a seguir:

Valor do Imposto Original	Multa por mês de atraso
até R\$ 99,99	R\$ 2,00
de R\$ 100,00 a R\$ 500,00	R\$ 4,00
mais de R\$ 500,00	R\$ 8,00

Por exemplo, se um imposto no valor de R\$ 200,00 estiver 3 meses atrasados, o valor da multa a ser paga será de R\$ 12,00 (3 x 4 Reais).

Escreva um programa completo que leia inicialmente o valor do imposto e o número de meses de atraso no pagamento (esses valores devem ser fornecidos pelo usuário do programa via o teclado). De acordo com o valor do imposto, o programa deve calcular a multa devida e imprimir uma mensagem para o usuário com o valor da multa e o novo valor do imposto, já acrescido da multa. Considerando o exemplo acima (imposto original de R\$ 200,00 e 3 meses de atraso), o seu programa deve imprimir a mensagem

Multa: R\$ 12,00

Imposto Total: R\$ 212,00

*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 – 11/12/06	Questão 2
Nome:	
Matrícula:	Turma

Escreva uma função em C para retirar o prefixo de uma cadeia de caracteres. A função deve receber como parâmetros de entrada uma cadeia de caracteres e o comprimento do prefixo que deve ser retirado. Essa função deve retornar uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo a cadeia original sem o prefixo. Por exemplo, se forem passados a cadeia “Puc-Rio” e o número 4, a função deve retornar uma nova cadeia “Rio”. O protótipo da função deve ser:

```
char* retira_prefixo (char* s, int n);
```

Se o comprimento do prefixo for maior do que o comprimento da cadeia original, a função deve retornar NULL.

INF 1620 – P4 – 11/12/06	Questão 3
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). Considere ainda que não existem linhas em branco no arquivo. Um exemplo desse formato é mostrado abaixo.

9010087-2	2.0	4.3	6.5
8820324-3	7.0	8.2	8.5
9210478-5	6.0	7.5	7.8
9020256-8	3.0	0.5	4.2

Escreva uma função que receba como parâmetros o número de matrícula de um aluno e o nome de um arquivo com as notas de uma disciplina no formato descrito acima, e retorne a média do aluno na disciplina. A média de um aluno é calculada pela fórmula  $(P1+P2+P3)/3$ . Caso o número de matrícula passado como parâmetro não seja encontrado no arquivo, a função deve retornar -1.0. Por exemplo, considerando um arquivo com o conteúdo apresentado acima, sua função deve retornar 7.9 caso o número de matrícula passado como parâmetro seja “8820324-3”. O protótipo da função deve ser:

```
float media (char* mat, char* nome_arquivo);
```

Obs.: Se não for possível abrir o arquivo de entrada, a função deve imprimir a mensagem “ERRO” e terminar a execução do programa.

INF 1620 – P4 – 11/12/06	Questão 4
Nome:	
Matrícula:	Turma

Considere a implementação de uma lista encadeada para armazenar as médias dos alunos de uma turma. O tipo que representa a lista é dado a seguir:

```
struct lista {
    char nome[81];
    float media;
    struct lista *prox;
};

typedef struct lista Lista;
```

Assumindo que a lista está em ordem alfabética pelo nome do aluno, escreva uma função para inserir um novo elemento na lista que mantenha essa ordenação (considere que todos os nomes estão em letras minúsculas). O protótipo da função deve ser:

```
Lista* insere (Lista* l, char* nome, float media);
```

INF 1620 – P4 – 11/12/06	Questão 5
Nome:	
Matrícula:	Turma

Considere uma *árvore binária de busca* que armazena dados de alunos de uma turma, usando a média do aluno como critério de ordenação. As médias dos alunos associados aos nós da sub-árvore à esquerda são menores que a média do aluno associado à raiz, e as médias dos alunos associados aos nós da sub-árvore à direita são maiores ou iguais à média do nó da raiz. O tipo que representa um nó da árvore é dado por:

```
struct arv {
    char nome[81];    /* nome */
    float media;     /* media obtida */
    struct arv* esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Pede-se:

- Escreva uma função que insira um novo aluno na estrutura da árvore, obedecendo o critério de ordenação. Essa função deve retornar o ponteiro para o nó raiz após a inserção e deve obedecer ao seguinte protótipo:

```
Arv* insere (Arv* a, char* nome, float media);
```

- Escreva uma função para imprimir os valores associados aos nós da árvore (nome e média) em ordem *decrecente*, apenas dos alunos *aprovados*. (para um aluno estar aprovado, sua média deve ser maior ou igual a 5.0). O protótipo dessa função é dado por:

```
void imprime_aprovados (Arv* a);
```

# 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.*