

INF 1620 – P4 - 06/12/05	Questão 1
Nome:	
Matrícula:	Turma

- (a) Escreva uma função que retorne uma aproximação do valor de  $\pi$ , de acordo com a fórmula de Leibniz:

$$\pi \approx 4 * \left( 1 - \frac{1}{3} + \frac{1}{5} - \frac{1}{7} + \frac{1}{9} - \frac{1}{11} \dots \right)$$

Isto é:

$$\pi \approx 4 * \sum_{i=0}^n \frac{-1^i}{2 * i + 1}$$

Essa função deve obedecer ao seguinte protótipo, em que  $n$  indica o número de termos que deve ser usado para avaliar o valor de  $\pi$ :

```
double pi (int n);
```

- (b) Escreva um programa completo que capture um valor inteiro e um valor de ponto flutuante, fornecidos pelo usuário via teclado, correspondentes aos números de termos a serem usados para o cálculo de  $\pi$  pela função do item *a* e ao raio de uma esfera, respectivamente. Com base nesses valores e utilizando a função do item *a* para o cálculo de  $\pi$ , seu programa deve imprimir para o usuário o volume e a área da superfície da esfera.

Fórmulas:

$$V = \frac{4}{3} \pi r^3$$

$$A = 4 \pi r^2$$

INF 1620 – P4 - 06/12/05	Questão 2
Nome:	
Matrícula:	Turma

Escreva uma função que receba como parâmetro uma string, e retorne uma nova string com todos os caracteres que sejam dígitos (caracteres entre '0' e '9') substituídos pelo caractere '?'. Por exemplo, se for passada como parâmetro a cadeia de caracteres "144-Leme", a função deve retornar a cadeia "???-Leme". A assinatura dessa função deve ser:

```
char* codifica (char* s);
```

*Notas:* A string passada como parâmetro não pode ser alterada. O espaço de memória para a nova string retornada por essa função deve ser alocado dinamicamente.

INF 1620 – P4 - 06/12/05	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 8.6
9210478-5 6.0 7.5 7.8
9020256-8 3.0 0.5 4.2
```

Escreva um programa completo que leia o arquivo “entrada.txt”, que contém as notas de uma disciplina no formato descrito acima, e escreva no arquivo texto “saida.txt” o número de matrícula de cada aluno com sua respectiva média  $((P1+P2+P3)/3)$  e um caractere indicando se o aluno foi aprovado ou reprovado na disciplina (use ‘A’ para indicar aprovação e ‘R’ para indicar reprovação). Considere que um aluno está aprovado se tiver média maior ou igual a 5.0. Para o exemplo acima, o programa deve criar um arquivo “saida.txt” com o seguinte conteúdo:

```
9010087-2 4.3 R
8820324-3 7.9 A
9210478-5 7.1 A
9020256-8 2.6 R
```

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.

INF 1620 – P4 - 06/12/05	Questão 4
Nome:	
Matrícula:	Turma

Considere uma lista encadeada que armazena strings. O tipo que representa cada nó da lista é dado por:

```
struct lista {
    char* str;          /* string armazenada */
    struct lista* prox; /* ponteiro para próximo elemento */
};
typedef struct lista Lista;
```

Escreva uma função que receba como parâmetros uma lista encadeada e uma string, e retire da lista todos os nós que armazenam uma string igual àquela passada como parâmetro. A função deve ter como valor de retorno o ponteiro para o primeiro nó da lista resultante. O protótipo dessa função é dado por:

```
Lista* retira (Lista* l, char* n);
```

INF 1620 – P4 - 06/12/05	Questão 5
Nome:	
Matrícula:	Turma

Considere a implementação de uma árvore binária de busca de números de ponto flutuante, em que os nós à esquerda da raiz armazenam valores menores que o da raiz, e os nós à direita armazenam valores maiores. Os elementos da árvore são definidos pela seguinte estrutura:

```
struct arv {
    float info;
    struct arv* esq;
    struct arv* dir;
};
typedef struct arv Arv;
```

Pede-se

- (a) Escreva uma função que tenha como valor de retorno a soma dos valores armazenados em todos os *nós folhas* de uma dada árvore. O protótipo dessa função deve ser:

```
float soma_folhas (Arv* a);
```

- (b) Escreva uma função para imprimir na tela do computador os valores armazenados na árvore em ordem decrescente. O protótipo dessa função é dado por:

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

*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.*