

INF 1620 – P1 - 17/04/04	Questão 1
Nome:	
Matrícula:	Turma

a) Implemente uma função que retorne uma aproximação do valor de π , de acordo com a Fórmula de Leibniz:

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

Ou seja:

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

Esta função deve obedecer ao protótipo abaixo, onde n indica o número de termos da série que devem ser usados para o cálculo do valor de π retornado.

```
float pi (int n);
```

b) Implemente um programa completo que capture via teclado o número de termos a ser utilizado na avaliação do valor π e, usando a função do item acima, calcule e imprima o valor de π calculado. Se o número de termos fornecido for menor do que um, o programa deve exibir uma mensagem de erro.

INF 1620 – P1 - 17/04/04	Questão 2
Nome:	
Matrícula:	Turma

A média aritmética, A , e a média harmônica, H , de um conjunto de valores são dadas pelas seguintes fórmulas:

$$A = \frac{1}{n} \sum v_i \qquad \frac{1}{H} = \frac{1}{n} \sum \frac{1}{v_i}$$

Implemente uma função que calcule a média aritmética e a média harmônica de um conjunto de valores armazenado num vetor. Esta função deve obedecer ao protótipo abaixo, recebendo como parâmetros o número de elementos do vetor, o endereço inicial do vetor e os endereços das variáveis onde os valores das médias, isto é, os valores de A e H , devem ser armazenados.

```
void medias (int n, float* v, float* pA, float* pH);
```

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 – P1 - 17/04/04	Questão 3
Nome:	
Matrícula:	Turma

Considere matrizes quadradas de dimensão $n \times n$. Essas matrizes podem ser representadas por vetores de dimensão $n * n$, onde os elementos da primeira linha da matriz ocupam as primeiras posições do vetor, seguidos dos elementos da segunda linha, e assim por diante, conforme ilustrado esquematicamente abaixo:

$$\begin{bmatrix} a & b & c \\ d & e & f \\ g & h & i \end{bmatrix} \longrightarrow [a \ b \ c \ d \ e \ f \ g \ h \ i]$$

Supondo que matrizes $n \times n$ de números reais são armazenadas em vetores, escreva uma função que crie dinamicamente uma matriz identidade. A função deve receber como parâmetro a dimensão da matriz, n , e retornar o ponteiro para a nova matriz criada, cujos elementos devem ser inicializados para se obter a matriz identidade. Esta função deve obedecer ao seguinte protótipo:

```
float* identidade (int n);
```

Obs: Numa matriz identidade, os elementos da diagonal principal têm valor 1.0 e os demais elementos têm valor 0.0.

INF 1620 – P1 - 17/04/04	Questão 4
Nome:	
Matrícula:	Turma

Implemente uma função que crie uma cadeia de caracteres inversa, trocando a ordem dos caracteres. A função deve receber como parâmetro de entrada uma cadeia de caracteres e retornar uma nova cadeia, cujo espaço de memória deve ser alocado pela função, contendo a cadeia original invertida: o primeiro caractere passa a ser o último, o segundo o penúltimo, e assim por diante. Assim, se for fornecido a cadeia “puc-rio”, deve-se retornar a cadeia “oir-cup”. A cadeia de caracteres original não pode ser alterada. A assinatura da função deve ser:

```
char* inverte (char* s);
```

INF 1620 – P1 - 17/04/04	Questão 5
Nome:	
Matrícula:	Turma

Considere um cadastro de alunos matriculados em uma disciplina, com as seguintes informações para cada aluno:

- Nome do aluno: com até 80 caracteres
- E-mail: representado por até 40 caracteres
- Número de matrícula: representado por um valor inteiro
- Notas obtidas em três provas, P1, P2, P3: representadas por valores reais

- (a) Defina uma estrutura em C, denominada `aluno`, que tenha os campos apropriados para guardar as informações de um aluno, conforme descrito acima.
- (b) Escreva uma função que receba como parâmetro um ponteiro para uma estrutura do tipo definido no item anterior e verifique se o dado aluno foi ou não aprovado. Para um aluno ser considerado aprovado, a média nas três notas deve ser maior ou igual a cinco (5.0) e nenhuma das notas pode ser menor do que três (3.0). Esta função deve retornar 1 se o aluno foi aprovado e 0 se o aluno não foi aprovado. A função deve obedecer ao seguinte protótipo:

```
int aprovado (struct aluno* a);
```

INF 1620 – P1 - 17/04/04	Questão 6
Nome:	
Matrícula:	Turma

Considere um cadastro de alunos matriculados em uma disciplina. Cada aluno é representado pelo seguinte tipo:

```
struct aluno {
    char nome[81];    /* nome do aluno */
    float p1;        /* nota na P1 */
    float p2;        /* nota na P2 */
    float p3;        /* nota na P3 */
};
```

- (a) Escreva uma função que, dados o nome e as três notas de um aluno, crie dinamicamente uma estrutura para representar esse aluno e retorne o ponteiro da área alocada. A nova estrutura deve ter seus campos inicializados com os valores passados como parâmetros. Essa função deve obedecer ao seguinte protótipo:

```
struct aluno* cria (char* nome, float p1, float p2, float p3);
```

- (b) Escreva uma função que receba como parâmetro um ponteiro para uma estrutura do tipo acima e verifique se o dado aluno foi ou não aprovado. Para um aluno ser considerado aprovado, a média nas três notas deve ser maior ou igual a cinco (5.0) e nenhuma das notas pode ser menor do que três (3.0). Esta função deve retornar 1 se o aluno foi aprovado e 0 se o aluno não foi aprovado. A função deve obedecer ao seguinte protótipo:

```
int aprovado (struct aluno* 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, ...);
```

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);
```

Não separe as folhas deste caderno.