

INF 1620 – P1 - 04/10/03	Questão 1
Nome:	
Matrícula:	Turma

a) Implemente uma função para calcular as raízes de uma equação do segundo grau do tipo $ax^2+bx+c=0$. O protótipo dessa função deve ser:

```
void raizes (float a, float b, float c, float* x1, float* x2);
```

onde a, b e c representam os coeficientes da equação e x1 e x2 são ponteiros para as raízes que devem ser calculadas. Considere que as raízes da equação sempre serão reais e distintas.

b) Implemente um programa completo que capture via teclado os três coeficientes de uma equação do segundo grau e, usando a função do item acima, calcule e imprima os valores das raízes correspondentes.

```
#include <math.h>
#include <stdio.h>

void raizes (float a, float b, float c, float *x1, float
*x2) {
    float d = sqrt(b*b - 4*a*c);
    *x1 = (-b - d)/(2*a);
    *x2 = (-b + d)/(2*a);
}

int main (void) {
    float a, b, c, x1, x2;
    printf("Entre com os coeficientes a, b, e c: ");
    scanf("%f %f %f", &a, &b, &c);
    raizes(a,b,c,&x1,&x2);
    printf("As raízes são: %f e %f\n", x1, x2);
    return 0;
}
```

Formulário:

Fórmula para cálculo das raízes: $x = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

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 - 04/10/03	Questão 2
Nome:	
Matrícula:	Turma

Dado um vetor de número reais, considere uma função, `acima_media`, para calcular o número de elementos do vetor cujos valores são maiores do que a média de todos os elementos. O programa abaixo ilustra uma utilização desta função, imprimindo o valor 3 que representa o número de valores acima da média (o valor da média para o vetor do exemplo abaixo vale 5.34).

```
#include <stdio.h>

int main (void) {
    int x;
    float v[5] = {4.5, 7.8, 6.5, 2.3, 5.6};
    /* calcula número de valores acima da média */
    x = acima_media (5, v);
    printf("Valores acima da media: %d\n", x);
    return 0;
}
```

Implemente a função `acima_media` de tal forma que o exemplo acima funcione adequadamente. A solução deve servir para qualquer vetor de reais, não podendo ser particularizada para o programa acima.

```
int acima_media(int n, float* v) {
    float m = 0.0;
    int i, res = 0;

    for (i = 0; i < n; i++)
        m = m + v[i];
    m = m/n;

    for (i = 0; i < n; i++)
        if (v[i] > m)
            res++;
    return res;
}
```

INF 1620 – P1 - 04/10/03	Questão 3
Nome:	
Matrícula:	Turma

Considere uma matriz de dimensão $m \times n$, onde m e n representam, respectivamente, o número de linhas e o número de colunas da matriz. Essa matriz pode ser representada por um vetor de dimensão $m \cdot n$, 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 a seguir:

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

Supondo que matrizes $m \times n$ de números reais são armazenadas em vetores, escreva uma função que, dada duas matrizes, retorne uma nova matriz, alocada dinamicamente pela função, que represente a soma das duas matrizes dadas. Isto é, se forem dadas as matrizes **A** e **B**, deve-se retornar uma nova matriz **C**, onde $C_{ij}=A_{ij}+B_{ij}$. O protótipo da função deve ser:

```
float* soma (int m, int n, float* a, float* b);
```

```
float* soma (int m, int n, float* a, float* b) {
    int i, j;
    float* c = (float*) malloc(m*n*sizeof(float));
    for (i = 0; i < m; i++)
        for (j = 0; j < n; j++)
            c[i*n+j] = a[i*n + j] + b[i*n + j];
    return c;
}
```

ou apenas:

```
float* soma (int m, int n, float* a, float* b) {
    int i, k;
    k = m*n;
    float* c = (float*) malloc(k*sizeof(float));
    for (i = 0; i < k; i++)
        c[i] = a[i] + b[i];
    return c;
}
```

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 - 04/10/03	Questão 4
Nome:	
Matrícula:	Turma

Implemente uma função que crie uma cadeia de caracteres codificada, trocando todas as letras por suas respectivas sucessoras no alfabeto. 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 codificada: ocorrências da letra *a* devem ser trocadas pela letra *b*, da letra *b* pela letra *c* e assim por diante; da letra *z* devem ser trocadas pela letra *a*. A mesma regra de trocas deve ser aplicada para as letras maiúsculas. Caracteres que não são letras devem permanecer inalterados. Assim, se for fornecido a cadeia “Urso-feroz”, deve-se retornar a cadeia “Vstp-gfspa”. A função não pode alterar o conteúdo da cadeia de caracteres original, e sua assinatura deve ser:

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

char* codifica(char* str) {
    char* s = (char*) malloc(strlen(str) + 1);
    int i;

    for (i = 0; str[i] != '\0'; i++)
        if (str[i] >= 'a' && str[i] < 'z')
            s[i] = str[i] + 1;
        else if (str[i] == 'z')
            s[i] = 'a';
        else if (str[i] >= 'A' && str[i] < 'Z')
            s[i] = str[i] + 1;
        else if (str[i] == 'Z')
            s[i] = 'A';
        else
            s[i] = str[i];

    s[i] = '\0';
    return s;
}

ou

char* codifica (char* str) {
    char* s = (char*) malloc(strlen(str) + 1);
    int i;

    for (i = 0; str[i] != '\0'; i++)
        if (str[i] >= 'a' && str[i] <= 'z')
            s[i] = 'a' + (str[i] - 'a' + 1)%('z' - 'a' + 1);
        else if (str[i] >= 'A' && str[i] <= 'Z')
            s[i] = 'A' + (str[i] - 'A' + 1)%('Z' - 'A' + 1);
        else
            s[i] = str[i];
    s[i] = '\0';
    return s;
}
```

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 - 04/10/03	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
- 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 imprima na tela do computador uma linha com o nome do aluno e outra linha com a média obtida nas três provas. Essa função deve seguir o seguinte protótipo:

```
void imprime (struct aluno* a);
```

```
a)
struct aluno {
    char nome[81];
    int mat;
    float p1;
    float p2;
    float p3;
};
```

```
b)
void imprime (struct aluno* a) {
    printf("%s\n", a->nome);
    printf("%f\n", (a->p1 + a->p2 + a->p3)/3);
}
```

RASCUNHO

Respostas nesta folha não serão consideradas.

Protótipos de funções que podem ser úteis:

stdio.h:

```
int scanf (char* formato, ...);  
int printf (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.