

Universidade de Itaúna - Ciência da Computação
Laboratório de Algoritmos e Estruturas de Dados I
Professor: Thiago Silva Vilela
Solução de Exercícios - Strings, vetores e matrizes

Exercício 1

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     float notas[10] = {5, 7, 2, 8, 6, 4, 4, 10, 1, 2};
6     float aumento;
7     int i;
8     printf("Digite a porcentagem de aumento: ");
9     scanf("%f", &aumento);
10    for (i = 0; i < 10; i++) {
11        notas[i] = notas[i] * (1.0 + aumento/100.0);
12        if (notas[i] > 10.0) {
13            notas[i] = 10.0;
14        }
15    }
16    printf("As novas notas sao: ");
17    for (i = 0; i < 10; i++) {
18        printf("%.2f ", notas[i]);
19    }
20    printf("\n");
21    return 0;
22 }
```

Exercício 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main(int argc, char *argv[]) {
6     srand(time(NULL));
7     int matriz[5][5];
8     int i, j, soma = 0;
9     //Inicializa a matriz
10    for (i = 0; i < 5; i++) {
11        for (j = 0; j < 5; j++) {
12            matriz[i][j] = rand()%5 + 5;
13        }
14    }
15    //Calcula a soma da diagonal. Como a matriz
16    //eh quadrada, precisamos somente de um for!
17    for (i = 0; i < 5; i++) {
18        soma += matriz[i][i];
19    }
20    //Imprime saida
21    for (i = 0; i < 5; i++) {
22        for (j = 0; j < 5; j++) {
23            printf("%d ", matriz[i][j]);
24        }
25        printf("\n");
26    }
27    printf("A soma da diagonal da matriz e: %d.\n", soma);
28    return 0;
29 }
```

Exercício 3

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(int argc, char *argv[]) {
5     float vetor1[5];
6     float vetor2[5];
7     float vetor_soma[5];
8     int i;
9     printf("Vetor 1: ");
10    for (i = 0; i < 5; i++) {
11        scanf("%f", &vetor1[i]);
12    }
13    printf("Vetor 2: ");
14    for (i = 0; i < 5; i++) {
15        scanf("%f", &vetor2[i]);
16    }
17    for (i = 0; i < 5; i++) {
18        vetor_soma[i] = vetor1[i] + vetor2[i];
19    }
20    printf("Vetor soma: ");
21    for (i = 0; i < 5; i++) {
22        printf("%.2f ", vetor_soma[i]);
23    }
24    return 0;
25 }
```

Exercício 4

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char palavra[101];
7     char segmento[101];
8     int tam_p, tam_s, i, j, eh_segmento = 0;
9     printf("String 1: ");
10    scanf("%s", palavra);
11    printf("String 2: ");
12    scanf("%s", segmento);
13    tam_p = strlen(palavra);
14    tam_s = strlen(segmento);
15    for (i = 0; i < tam_p; i++) {
16        // A primeira letra da segunda string deve ser encontrada
17        // na primeira string para que a segunda seja um segmento
18        // da primeira!
19        if (palavra[i] == segmento[0]) {
20            // Se encontramos a primeira letra do segmento na palavra, eh
21            // possivel
22            // que o segmento esteja na palavra. Vamos testar o restante do
23            // segmento.
24            eh_segmento = 1;
25            for (j = 1; j < tam_s; j++) {
26                // Se a primeira palavra acabou (i+j >= tam_p) ou se alguma letra
27                // do segmento
28                // eh diferente do que esperavamos, entao nao encontramos o
29                // segmento na palavra.
30                if (i+j >= tam_p || palavra[i+j] != segmento[j]) {
31                    eh_segmento = 0;
32                    break;
33                }
34            }
35            // Se aqui eh_segmento=1, entao quer dizer que nossa checagem
36            // anterior
37            // foi correta, e encontramos o segmento na palavra!
38            if (eh_segmento == 1) {
39                printf("A segunda string esta contida na primeira!\n");
40                break;
41            }
42        }
43    }
44    if (!eh_segmento) {
45        printf("A segunda string nao esta contida na primeira!\n");
46    }
47    return 0;
48 }
```

Exercício 5

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char string[101];
7     int tam, i, j;
8     printf("Forneca uma string com espacos em branco:");
9     gets(string);
10    tam = strlen(string);
11    j = 0;
12    //Remove espacos em branco
13    for (i = 0; i < tam; i++) {
14        if (string[i] != ' ') {
15            string[j++] = string[i];
16        }
17    }
18    string[j] = '\0';
19    //Conversao de letras maiusculas para minusculas.
20    //Basta adicionar 32 ao codigo ASCII de um caracter
21    //maiusculo e ele se tornara minusculo.
22    tam = strlen(string);
23    for (i = 0; i < tam; i++) {
24        if (string[i] >= 65 && string[i] <= 90) {
25            string[i] += 32;
26        }
27    }
28    printf("String resultante: %s\n", string);
29    return 0;
30 }
```

Exercício 6

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 /**
4  * Para checar se uma matriz eh de permutacao precisamos garantir que:
5  * - Ela seja formada somente de 0s e 1s;
6  * - A soma de todas as suas linhas seja 1;
7  * - A soma de todas as suas colunas seja 1.
8  */
9 int main(int argc, char *argv[]) {
10     int matriz[4][4];
11     int i, j, soma;
12     int eh_permutacao;
13     printf("Entre com os elementos da matriz:\n");
14     for (i = 0; i < 4; i++) {
15         for (j = 0; j < 4; j++) {
16             scanf("%d", &matriz[i][j]);
17         }
18     }
19     // Inicialmente, vamos considerar que a matriz eh de permutacao
20     eh_permutacao = 1;
21     // Verifica se as linhas da matriz tem soma 1. Por eficiencia,
22     // vamos aprovaritar esse momento para verificar tambem se todos os
23     // elementos sao 0 e 1. Caso a soma de alguma linha nao seja 1 ou caso
24     // algum elemento nao seja 1 ou 0, vamos marcar eh_permutacao como
25     // falso, ou seja, eh_permutacao= 0.
26     for (i = 0; i < 4; i++) {
27         soma = 0;
28         for (j = 0; j < 4; j++) {
29             soma += matriz[i][j];
30             if (matriz[i][j] != 1 && matriz[i][j] != 0) {
31                 eh_permutacao = 0;
32                 break;
33             }
34         }
35         if (!eh_permutacao || soma > 1) {
36             eh_permutacao = 0;
37             break;
38         }
39     }
40     // Verifica se as colunas da matriz tem soma 1. Nesse momento
41     // ja verificamos se os elementos sao todos 0 e 1, entao nao
42     // precisamos repetir tal verificacao.
43     for (i = 0; i < 4; i++) {
44         soma = 0;
45         for (j = 0; j < 4; j++) {
46             soma += matriz[j][i];
47         }
48         if (soma > 1) {
49             eh_permutacao = 0;
50             break;
51         }
52     }
53     if (eh_permutacao) {
54         printf("A matriz fornecida e de permutacao!\n");
55     } else {
56         printf("A matriz fornecida nao e de permutacao!\n");
57     }
58     return 0;
59 }
```