

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     int vetor1[10];
6     int vetor2[10];
7     int i;
8     int resultado = 0;
9     printf("Vetor 1: ");
10    for (i = 0; i < 10; i++) {
11        scanf("%d", &vetor1[i]);
12    }
13    printf("Vetor 2: ");
14    for (i = 0; i < 10; i++) {
15        scanf("%d", &vetor2[i]);
16    }
17    for (i = 0; i < 10; i++) {
18        resultado += vetor1[i] * vetor2[i];
19    }
20    printf("Resultado: %d\n", resultado);
21    return 0;
22 }
```

Exercício 2

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char palavra[103]; //100 caracteres + \0 + ay
7     char c;
8     int i, tam;
9     printf("String: ");
10    scanf("%s", palavra);
11    tam = strlen(palavra);
12    c = palavra[0];
13    for (i = 0; i < tam; i++) {
14        palavra[i] = palavra[i+1];
15    }
16    palavra[tam-1] = c;
17    palavra[tam] = '\0';
18    strcat(palavra, "ay");
19    printf("Palavra resultante: %s\n", palavra);
20    return 0;
21 }
```

Exercício 3

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char c1[301], c2[301], c3[301];
7     printf("Entre com 3 palavras: ");
8     scanf("%s %s %s", c1, c2, c3);
9     if (strcmp(c1, c2) > 0 && strcmp(c2, c3) > 0) {
10         strcat(c1, c2);
11         strcat(c1, c3);
12         printf("Resultado: %s\n", c1);
13     } else if (strcmp(c1, c3) > 0 && strcmp(c3, c2) > 0) {
14         strcat(c1, c3);
15         strcat(c1, c2);
16         printf("Resultado: %s\n", c1);
17     } else if (strcmp(c2, c1) > 0 && strcmp(c1, c3) > 0) {
18         strcat(c2, c1);
19         strcat(c2, c3);
20         printf("Resultado: %s\n", c2);
21     } else if (strcmp(c2, c3) > 0 && strcmp(c3, c1) > 0) {
22         strcat(c2, c3);
23         strcat(c2, c1);
24         printf("Resultado: %s\n", c2);
25     } else if (strcmp(c3, c1) > 0 && strcmp(c1, c2) > 0) {
26         strcat(c3, c1);
27         strcat(c3, c2);
28         printf("Resultado: %s\n", c3);
29     } else if (strcmp(c3, c2) > 0 && strcmp(c2, c1) > 0) {
30         strcat(c3, c2);
31         strcat(c3, c1);
32         printf("Resultado: %s\n", c3);
33     }
34     return 0;
35 }
```

Exercício 4

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int main(int argc, char *argv[]) {
6     char all_words[100][21];
7     char concat[2001] = "";
8     int n, i;
9     printf("Tamanho do vetor: ");
10    scanf("%d", &n);
11    printf("Forneca as palavras:\n");
12    for (i = 0; i < n; i++) {
13        scanf("%s", all_words[i]);
14    }
15    for (i = 0; i < n; i++) {
16        strcat(concat, all_words[i]);
17    }
18    printf("A palavra concatenada e: %s\n", concat);
19    return 0;
20 }
```

Exercício 5

```
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 repetidos[101];
9     int i, j, tem_repetidos = 0;
10    printf("Matriz gerada:\n");
11    for (i = 0; i < 5; i++) {
12        for (j = 0; j < 5; j++) {
13            matriz[i][j] = rand()%100 + 1;
14            printf("%3d ", matriz[i][j]);
15        }
16        printf("\n");
17    }
18    //Vamos usar o vetor repetidos como um contador.
19    //Inicialmente, precisamos inicializa-lo com 0.
20    for (i = 0; i <= 100; i++) {
21        repetidos[i] = 0;
22    }
23    //Sempre que um numero x e visto na matriz,
24    //incrementamos a posicao x do vetor repetidos.
25    for (i = 0; i < 5; i++) {
26        for (j = 0; j < 5; j++) {
27            repetidos[matriz[i][j]]++;
28        }
29    }
30    //Se alguma posicao do vetor repetidos e maior que 1,
31    //temos elementos repetidos.
32    for (i = 1; i <= 100; i++) {
33        if (repetidos[i] > 1) {
34            printf("A matriz gerada possui elementos repetidos!\n");
35            tem_repetidos = 1;
36            break;
37        }
38    }
39    if (!tem_repetidos) {
40        printf("A matriz gerada nao possui elementos repetidos!\n");
41    }
42    return 0;
43 }
```