

Resposta da 6ª Lista de Exercícios

Assunto

Resposta pergunta 1:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #define MAXTAM 100
4 int main(){
5     char frase[MAXTAM];
6     char fraseMaiuscula[MAXTAM];
7     char fraseMinuscula[MAXTAM];
8     int i = 0;
9     printf("Digite uma palavra / frase:");
10    fgets(frase, 100, stdin); //fgets le ate o \n
11    while(frase[i] != '\0'){
12        fraseMaiuscula[i] = toupper(frase[i]);
13        fraseMinuscula[i] = tolower(frase[i]);
14        i++;
15    }
16    printf("Frase Maiuscula: %s\n", fraseMaiuscula);
17    printf("Frase Minuscula: %s\n", fraseMinuscula);
18    return 0;
19 }
```

Listing 1: Resposta do exercício 1 codificado na linguagem C

Resposta pergunta 2:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #define MAXTAM 100
4 int main(){
5     char palavra[MAXTAM];
6     char c;
7     int i = 0, cont = 0;
8     printf("Digite uma palavra: ");
9     scanf("%s", palavra); //Le apenas uma palavra
10    getchar(); //Le o "pula linha"
11    printf("Digite um caracter: ");
12    scanf("%c", &c);
13    c = tolower(c);
14    while(palavra[i] != '\0'){
15        if(tolower(palavra[i]) == c)
16            cont++;
17        i++;
18    }
19    printf("Na palavra '%s' o caracter '%c' aparece %d vezes", palavra, c, cont)
20    ;
21    return 0;
22 }
```

Listing 2: Resposta do exercício 2 codificado na linguagem C

Resposta pergunta 3:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #define MAXTAM 100
4 int main() {
5     char palavra[MAXTAM], palavra_final[MAXTAM];
6     char c;
7     int i = 0, j = 0;
8     printf("Digite uma palavra: ");
9     scanf("%s", palavra);
10    getchar();
11    printf("Digite um caracter: ");
12    scanf("%c", &c);
13    c = tolower(c);
14    while(palavra[i] != '\0'){
15        if(tolower(palavra[i]) != c){
16            palavra_final[j] = palavra[i];
17            j++;
18        }
19        i++;
20    }
21    palavra_final[j] = '\0';
22    printf("A palavra '%s' sem o caracter '%c' eh '%s'", palavra, c,
23        palavra_final);
24    return 0;
25 }
```

Listing 3: Resposta do exercício 3 codificado na linguagem C

Resposta pergunta 4:

```
1 #include <stdio.h>
2 #include <ctype.h>
3 #define MAXTAM 100
4 #define MAX_CHAR 256
5 int main() {
6     char palavra[MAXTAM];
7     int num_char[MAX_CHAR] = {0};
8     char tmp;
9     int i = 0;
10    printf("Digite uma palavra: ");
11    scanf("%s", palavra);
12    while(palavra[i] != '\0'){
13        tmp = tolower(palavra[i]);
14        num_char[(int)tmp]++;
15        i++;
16    }
17    printf("Caracteres repetidos:\n");
18    for(i=0; i<MAX_CHAR; i++) {
19        if(num_char[i] > 1){
20            printf("%c\n", i);
21        }
22    }
23    return 0;
24 }
```

24 }

Listing 4: Resposta do exercício 4 codificado na linguagem C

Resposta pergunta 5:

```
#include <stdio.h>
2 #include <ctype.h>
#define MAXTAM 100
4 #define MAX_CHAR 256
int main() {
6 char palavra[MAXTAM], palavra_final[MAXTAM];
  int num_char[MAX_CHAR] = {0};
8 char tmp;
  int i = 0, j = 0;
10 printf("Digite uma palavra: ");
  scanf("%s", palavra);
12 while(palavra[i] != '\0'){
    tmp = tolower(palavra[i]);
14    num_char[(int)tmp]++;
    i++;
16 }
  i = 0;
18 while(palavra[i] != '\0'){
    if(num_char[tolower(palavra[i])] == 1){
20        palavra_final[j] = palavra[i];
        j++;
22    }
    i++;
24 }
  palavra_final[j] = '\0';
26 printf("A palavra '%s' sem os caracteres repetidos eh '%s'.\n", palavra,
  palavra_final);
  return 0;
28 }
```

Listing 5: Resposta do exercício 5 codificado na linguagem C

Resposta pergunta 6:

```
#include <stdio.h>
2 #include <ctype.h>
#define MAXTAM 100
4 int main() {
  char nome[MAXTAM], nome_abrev[MAXTAM];
6 char c;
  int i = 0;
8 printf("Digite um nome completo: ");
  scanf("%s", nome); //Le o primeiro nome
10 c = getchar(); //Le o caracter em seguida
  while(c != 10) { //Caracter de numero 10 eh o '\n'
12     nome_abrev[i++] = toupper(nome[0]);
     nome_abrev[i++] = '.';
14     nome_abrev[i++] = ' ';
     scanf("%s", nome); //Le o proximo nome
16     c = getchar();
```

```

18     }
    nome_abrev[i] = '\0';
    printf("%s, %s\n", nome, nome_abrev);
20     return 0;
}

```

Listing 6: Resposta do exercício 6 codificado na linguagem C

Resposta pergunta 7:

```

1 #include <stdio.h>
#include <ctype.h>
3 #include <string.h>
#define MAXTAM 100
5 int main() {
    char frase[MAXTAM];
7     int i, tam, flag = 1;
    printf("Digite uma frase: ");
9     fgets(frase, MAXTAM, stdin);
    tam = strlen(frase) - 2;
11    for (i = 0; i < (tam/2); i++){
        if (toupper(frase[i]) != toupper(frase[tam-i])){
13            flag = 0;
            break;
15        }
    }
17    if (flag){
        printf("A frase eh um palindromo.\n");
19    } else{
        printf("A frase nao eh um palindromo.\n");
21    }
    return 0;
23 }

```

Listing 7: Resposta do exercício 7 codificado na linguagem C

Resposta pergunta 8:

```

1 #include <stdio.h>
#include <string.h>
3 #define MAXTAM 100

5 int main() {
    char palavra[MAXTAM], chave;
7     int i, len;

9     printf("Digite uma chave: ");
    scanf("%c", &chave);
11    printf("Digite uma palavra: ");
    scanf("%s", palavra);

13    if (chave <= 'Z') chave = chave - 'A' + 1;
15    else chave = chave - 'a' + 1;
    len = strlen(palavra);

17    for (i = 0; i < len; i++) {

```

```

19     if (palavra[i] >= 65 && palavra[i] <= 90) {
20         palavra[i] = (palavra[i] - 'A' + chave) % 26 + 'A';
21     } else if (palavra[i] >= 97 && palavra[i] <= 122) {
22         palavra[i] = (palavra[i] - 'a' + chave) % 26 + 'a';
23     }
24 }
25 printf("Palavra criptografada: '%s'\n", palavra);
26 return 0;
27 }

```

Listing 8: Resposta do exercício 8 codificado na linguagem C

Resposta pergunta 9:

```

1 #include <stdio.h>
2 #include <string.h>
3 #define MAXTAM 100
4
5 int main() {
6     char string[MAXTAM], novaString[MAXTAM];
7     int i, len;
8
9     printf("Digite uma string: ");
10    fgets(string, MAXTAM, stdin);
11
12    len = strlen(string);
13
14    for (i = 0; i < len; i++)
15        if(toupper(string[i]) == 'A' || toupper(string[i]) == 'E' || toupper(
16            string[i]) == 'I' || toupper(string[i]) == 'O' || toupper(string[i]) == 'U'
17        ))
18            novaString[i] = '*';
19        else
20            novaString[i] = string[i];
21    novaString[i] = '\0';
22
23    printf("Frase original: %s", string);
24    printf("Frase sem vogais: %s", novaString);
25    return 0;
26 }

```

Listing 9: Resposta do exercício 9 codificado na linguagem C

Resposta pergunta 10:

```

1 #include <stdio.h>
2
3 #define MAXTAM 100
4 #define LETRAS 26
5
6 int main() {
7     char tmp, string1[MAXTAM], string2[MAXTAM];
8     int i, contChar1[LETRAS] = {0}, contChar2[LETRAS] = {0};
9
10    printf("Digite a primeira frase: ");
11    fgets(string1, MAXTAM, stdin);

```

```

12     printf("Digite a segunda frase: ");
13     fgets(string2 ,MAXTAM,stdin);
14
15     i=0;
16     while(string1[i] != '\0'){
17         tmp = tolower(string1[i]) - 'a';
18         if(tmp>=0 && tmp<LETRAS)
19             contChar1 [(int)tmp]++;
20         i++;
21     }
22
23     i=0;
24     while(string2[i] != '\0'){
25         tmp = tolower(string2[i]) - 'a';
26         if(tmp>=0 && tmp<LETRAS)
27             contChar2 [(int)tmp]++;
28         i++;
29     }
30
31     for(i=0; i<LETRAS && contChar1[i] == contChar2[i]; i++);
32
33     if(i==LETRAS)
34         printf("As frases sao anagramas\n");
35     else
36         printf("As frases nao sao anagramas\n");
37
38     return 0;
39 }

```

Listing 10: Resposta do exercício 10 codificado na linguagem C

Resposta pergunta 11:

```

1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAXTAM 100
5 #define MIN(a,b) a < b ? a : b
6
7 int main() {
8     char string1[MAXTAM], string2[MAXTAM];
9     int i, len1, len2, menorTamanho, res = 0;
10
11     printf("Digite a primeira frase: ");
12     fgets(string1, MAXTAM, stdin);
13
14     printf("Digite a segunda frase: ");
15     fgets(string2, MAXTAM, stdin);
16
17     len1 = strlen(string1);
18     len2 = strlen(string2);
19     menorTamanho = MIN(len1, len2);
20
21     for (i = 0; i < menorTamanho && res == 0; i++) {

```

```

23     if (tolower(string1[i]) < tolower(string2[i]))
        res = 1;
25     if (tolower(string1[i]) > tolower(string2[i]))
        res = 2;
27 }
28 if (res == 0 && len1 < len2)
29     res = 1;
30 if (res == 0 && len1 > len2)
31     res = 2;
32
33 if(res == 0)
    printf("As frases sao iguais\n");
35 else if(res == 1)
    printf("A primeira frase tem menor ordem lexicografica\n");
37 else
    printf("A segunda frase tem menor ordem lexicografica\n");
39
40 return 0;
41 }

```

Listing 11: Resposta do exercício 11 codificado na linguagem C

Resposta pergunta 12:

```

1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAXTAM 100
5
6 int main() {
7     char string1 [MAXTAM], string2 [MAXTAM];
8     int i, len1, len2;
9
10
11     printf("Digite a primeira string: ");
12     fgets(string1, MAXTAM, stdin);
13
14     printf("Digite a segunda string: ");
15     fgets(string2, MAXTAM, stdin);
16
17     len1 = strlen(string1);
18     len2 = strlen(string2);
19
20     for (i = len1-1; i < len1+len2-1 && i < MAXTAM; i++) {
21         string1[i] = string2[i-len1+1];
22     }
23     if(len1+len2 < MAXTAM) string1[len1+len2-1] = '\0';
24
25     printf("A string concatenada fica: %s", string1);
26
27     return 0;
28 }

```

Listing 12: Resposta do exercício 12 codificado na linguagem C

Resposta pergunta 13:

```
1 #include <stdio.h>
2 #include <string.h>
3
4 #define MAXTAM 100
5
6 int main() {
7     char string[MAXTAM];
8     int i = 0, pos, tamanho, maior = 0, menor = 0, contMenor = 0, contMaior =
9     0;
10    int posMaior[MAXTAM] = {0}, posMenor[MAXTAM] = {0}, len;
11
12    printf("Digite a frase: ");
13    fgets(string, MAXTAM, stdin);
14
15    len = strlen(string);
16
17    while (tolower(string[i]) >= 'a' && tolower(string[i]) <= 'z' && i < len)
18    {
19        maior++;
20        menor++;
21        i++;
22    }
23
24    if (maior > 0) {
25        contMaior++;
26        contMenor++;
27    }
28
29    for (; i < len; i++) {
30        pos = i;
31        while (tolower(string[i]) >= 'a' && tolower(string[i]) <= 'z' && i <
32        len) {
33            i++;
34        }
35        tamanho = i - pos;
36        if (tamanho == 0);
37        else {
38            if (tamanho > maior) {
39                maior = tamanho;
40                contMaior = 0;
41                posMaior[contMaior++] = pos;
42            } else if (tamanho == maior)
43                posMaior[contMaior++] = pos;
44
45            if (tamanho < menor) {
46                menor = tamanho;
47                contMenor = 0;
48                posMenor[contMenor++] = pos;
49            } else if (tamanho == menor)
50                posMenor[contMenor++] = pos;
51        }
52    }
53 }
```



```

50     }
52     printf("Majores palavras: ");
54     for (i = 0; i < contMaior; i++) {
55         while (tolower(string[posMaior[i]]) >= 'a' && tolower(string[posMaior[
56 i]]) <= 'z') {
57             printf("%c", string[posMaior[i]]);
58             posMaior[i]++;
59         }
60         if (i + 1 != contMaior) printf(", ");
61     }
62     printf("\n");
63
64     printf("Menores palavras: ");
65     for (i = 0; i < contMenor; i++) {
66         while (tolower(string[posMenor[i]]) >= 'a' && tolower(string[posMenor[
67 i]]) <= 'z') {
68             printf("%c", string[posMenor[i]]);
69             posMenor[i]++;
70         }
71         if (i + 1 != contMenor) printf(", ");
72     }
73     printf("\n");
74
75     return 0;
76 }

```

Listing 13: Resposta do exerc cio 13 codificado na linguagem C

Resposta pergunta 14:

```

#include <stdio.h>
2 #include <string.h>
#define MAXTAM 500
4
5 int main() {
6     char string[MAXTAM];
7     int i, j, len, palavrasIndex[MAXTAM], contPalavras = 0, aux;
8
9
10    printf("Digite o texto: ");
11    fgets(string, MAXTAM, stdin);
12
13    len = strlen(string);
14
15    palavrasIndex[contPalavras++] = 0;
16    for (i = 0; i < len; i++) {
17        if (string[i] == ' ') {
18            string[i] = '\0';
19            palavrasIndex[contPalavras++] = i + 1;
20        } else
21            string[i] = tolower(string[i]);
22    }
23    string[len - 1] = '\0';

```

```

24
26 for (i = 0; i < contPalavras; i++) {
28     for (j = i + 1; j < contPalavras; j++) {
29         if (strcmp(&string[palavrasIndex[i]], &string[palavrasIndex[j]]) >
30             0) {
31             aux = palavrasIndex[i];
32             palavrasIndex[i] = palavrasIndex[j];
33             palavrasIndex[j] = aux;
34         }
35     }
36 }
37
38 printf("%s\n",&string[ palavrasIndex [0]]);
39 for (i = 1; i < contPalavras; i++) {
40     if (strcmp(&string[ palavrasIndex [i]], &string[ palavrasIndex [i - 1]])
41         != 0)
42         printf("%s\n", &string[ palavrasIndex [i]]);
43 }
44 return 0;
45 }

```

Listing 14: Resposta do exercício 14 codificado na linguagem C

Resposta pergunta 15:

```

#include <stdio.h>
2 #include <string.h>
3 #define MAXTAM 500
4 #define TRUE 1
5 #define FALSE 0
6
7 int main() {
8     char string1[MAXTAM], string2[MAXTAM];
9     int len, len1, len2, i, j, r;
10
11
12     printf("Digite a primeira string: ");
13     fgets(string1, MAXTAM, stdin);
14
15     printf("Digite a segunda string: ");
16     fgets(string2, MAXTAM, stdin);
17
18     len1 = strlen(string1);
19     len2 = strlen(string2);
20     len2--;
21     string2[len2] = '\0';
22     len = len1 - len2;
23
24     r = FALSE;
25     for (i = 0; i <= len; i++) {
26         for (j = 0; j < len2 && string2[j] == string1[i + j]; j++);
27         if (j == len2) r = TRUE;
28     }
29 }

```

```

28     }
30     if(!r)
31         printf("A segunda string nao esta contida na primeira\n");
32     else
33         printf("A segunda string esta contida na primeira\n");
34
35     return 0;
36 }

```

Listing 15: Resposta do exercício 15 codificado na linguagem C

Resposta pergunta 16:

```

#include <stdio.h>
#include <string.h>

#define MAXTAM 200
#define MIN(a,b) a < b ? a : b

int main() {
    char texto[MAXTAM], textoSeparado[MAXTAM][MAXTAM][MAXTAM] = {0}, aux[
MAXTAM];
    int len, i = 0, j = 0, k = 0, m, rows, palavras[MAXTAM] = {0},
totalPalavras = 0;

    printf(" Digite o texto: ");
    fgets(texto, MAXTAM, stdin);

    len = strlen(texto);

    for (m = 0; m < len; m++) {
        if (texto[m] == ' ') {
            j++;
            palavras[i]++;
            k = 0;
        } else if (texto[m] == '.') {
            palavras[i]++;
            i++;
            j = 0;
            k = 0;
            if (texto[m + 1] == ' ')
                m++;
        } else {
            textoSeparado[i][j][k] = tolower(texto[m]);
            k++;
        }
    }

    rows = i;
    printf("O texto tem %d frases\n", rows);

    for (i = 0; i < MAXTAM && palavras[i] != 0; i++) {
        totalPalavras += palavras[i];
    }
}

```

```

40 printf("O texto tem %d palavras\n", totalPalavras);
42 for (i = 0; i < rows; i++) {
44     for (j = 0; j < palavras[i]; j++) {
46         for (k = j+1; k < palavras[i]; k++) {
48             if (strcmp(textoSeparado[i][j], textoSeparado[i][k]) > 0) {
50                 strcpy(aux, textoSeparado[i][j]);
52                 strcpy(textoSeparado[i][j], textoSeparado[i][k]);
54                 strcpy(textoSeparado[i][k], aux);
56             }
58         }
60     }
62     for (i = 0; i < rows; i++){
64         printf("Palavras da frase %d em ordem: ", i);
66         for (j=0; j<palavras[i]; j++){
68             printf("%s ", textoSeparado[i][j]);
70         }
72         printf("\n");
74     }
76     return 0;
78 }

```

Listing 16: Resposta do exercício 16 codificado na linguagem C