

Lista 9

1 - Relacione as colunas

Função para copiar uma string == strcpy(destino, origem)
Função para concatenar duas strings == strcat(destino, origem)
Função para encontrar o tamanho de uma string == strlen(string)
Função para comparar duas strings == strcmp(string1, string2)
Função para converter uma string para maiúsculas ==strupr(string)
Função para converter uma string para minúsculas ==strlwr(string)

2 -Implementar um programa que leia uma frase do usuário e execute as seguintes tarefas:

- Contar o número de **palavras** na frase.
- Contar o número de **caracteres** na frase.
- Exibir a frase com todas as palavras em maiúsculas.
- Exibir a frase com todas as palavras em minúsculas.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main(){
    char frase[61];
    int palavra = 0, i, cont = 0;

    printf("Digite uma frase: ");
    gets(frase);
    printf("\nSua frase e: %s\n", frase);

    printf("\nSua frase tem %i caracteres.\n", strlen(frase));

    for (i = 0; frase[i] != '\0'; i++) {
        if (isspace(frase[i])) {
            palavra = 0;
        } else if (!palavra) {
            palavra = 1;
            cont++;
        }
    }
    if(cont != 0){
        printf("\nSua frase tem %i palavras.\n", cont);
    }
}
```

```

   strupr(frase);
    printf("\nSua frase em maiusculas: %s\n", frase);

    strlwr(frase);
    printf("\nSua frase em minusculas: %s\n\n", frase);

    return 0;
}

```

3 - Implementar um programa que leia uma string do usuário e execute as seguintes tarefas:

- Inverter a ordem dos caracteres da string.
- Verificar se a string é um palíndromo (uma palavra, frase, número ou outra sequência de caracteres que lê da mesma forma tanto da esquerda para a direita quanto da direita para a esquerda).

```

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

```

```

int main(){
    char stri[81];
    printf("\nDigite uma string: ");
    gets(stri);
    int i, tam = strlen(stri);
    bool palindromo;

    for(i = 0; i < tam / 2; i++){
        if(stri[i] != stri[tam - i - 1]){
            palindromo = false;
        }else{
            palindromo = true;
        }
    }

    if (palindromo == true) {
        printf("\nA string '%s' e um palindromo.\n", stri);
    } else {
        printf("\nA string '%s' nao e um palindromo.\n", stri);
    }

    for(i = 0; i < tam / 2; i++){
        char cont = stri[i];
        stri[i] = stri[tam - i - 1];
        stri[tam - i - 1] = cont;
    }
}

```

```

    }
    printf("\nA string invertida e: '%s'\n", stri);

    return 0;
}

```

4 - Implementar um programa que leia uma string do usuário e execute as seguintes tarefas:

- Remover os espaços da string.
- Substituir todas as ocorrências de uma determinada letra por outra '*'.

```

#include <stdio.h>
#include <stdlib.h>
#include <ctype.h>
#include <string.h>
int main(){
    char algo[61], letra;
    int i, cont = 0;
    printf("Digite uma string: ");
    gets(algo);
    printf("\nVoce digitou: %s\n", algo);

    int tam = strlen(algo);
    printf("%s\n", algo);
    for (i = 0; i < tam; i++) {
        if (algo[i] == ' ') continue;
        algo[cont] = algo[i];
        cont++;
    }
    algo[cont] = '\0';
    printf("\nA string sem os espacos: %s\n", algo);

    printf("\nDigite a letra a ser substituida: ");
    scanf("%c", &letra);

    for (i = 0; i < tam; i++) {
        if (algo[i] == letra) {
            algo[i] = '*';
        }
    }
    printf("\nA string sem a letra '%c' fica: %s\n\n", letra, algo);

    return 0;
}

```

5 - Implementar um programa que leia uma string do usuário e execute as seguintes tarefas:

- Remover todas as vogais da string.
- Contar o número de vogais na string.

```
#include <stdio.h>
#include <ctype.h>
#include <string.h>
int main(){
    char frase[61];
    printf("Digite uma frase: ");
    gets(frase);
    printf("\nSua frase e: %s", frase);

    frase[strcspn(frase, "\n")] = '\0';
    int tam = strlen(frase), i, z = 0, vogais = 0;

    for (i = 0; i < tam; i++) {
        if (frase[i] == 'A' || frase[i] == 'E' || frase[i] == 'I' || frase[i] == 'O' || frase[i] ==
'U' || frase[i] == 'a' || frase[i] == 'e' || frase[i] == 'i' || frase[i] == 'o' || frase[i] == 'u')
        {
            vogais++;
        }
    }
    printf("\n\nA frase tem %d vogais.\n", vogais);

    for (i = 0; i < tam; i++) {
        if (frase[i] != 'A' && frase[i] != 'E' && frase[i] != 'I' && frase[i] != 'O' &&
frase[i] != 'U' && frase[i] != 'a' && frase[i] != 'e' && frase[i] != 'i' && frase[i] != 'o'
&& frase[i] != 'u') {
            frase[z++] = frase[i];
        }
    }
    frase[z] = '\0';
    printf("\nA frase sem as vogais fica: %s\n\n", frase);

    return 0;
}
```