

Strings

Um string nada mais é que uma seqüência especial de caracteres. Em C, eles são colocadas entre ". Assim, "oba", "teste\n" (que usamos muito no printf) são exemplos de string.

Agora não confunda: 'x' é o CHARACTER x, enquanto que "x" é o STRING x. E qual a diferença? Já vamos ver.

O C, infelizmente, não tem um tipo string pré-definido. Ou seja, ele tem que representar string de outra forma. E como ele representa? Como um vetor de caracteres terminado pelo caracter '\0'. Sendo assim, a única diferença entre um vetor de caracteres e uma string é a obrigatoriedade do '\0' no final da string.

Agora sim, podemos ver a diferença entre 'x' e "x". Quando escrevemos "x" o compilador, na verdade, cria um vetor com 2 caracteres: 'x' e '\0'. Por isso você não deve fazer confusão.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Lendo Strings

```
char str[100];  
  
// Solicita que o usuário insira uma string  
printf("Digite uma string (máx 99 caracteres): ");  
fgets(str, sizeof(str), stdin);
```



Acessando caractere por caractere

```
char str[] = "meu string";  
char letra;  
  
printf("%c\n",str[2]); /*imprime o 3o caracter*/  
  
letra = str[8];  
  
printf("%c\n",letra); /*imprime o 9o caracter*/
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

strlen - Calcula o comprimento de uma string

A função strlen retorna o número de caracteres em uma string, excluindo o caractere nulo \0.

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

int main() {
    char str[] = "Hello, world!";
    int length = strlen(str);
    printf("Comprimento da string: %d\n", length);
    return 0;
}
```



JESUÍTAS BRASIL



UNISINOS

Somos infinitas possibilidades

strcpy - Copia uma string para outra

A função strcpy copia uma string para outra.

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

int main() {
    char source[] = "Hello, world!";
    char destination[50];
    strcpy(destination, source);
    printf("Destino: %s\n", destination);
    return 0;
}
```



strncpy - Copia os primeiros n caracteres de uma string para outra

A função strncpy copia os primeiros n caracteres de uma string para outra. É mais segura que strcpy porque permite especificar o tamanho do destino.

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

int main() {
    char source[] = "Hello, world!";
    char destination[50];
    strncpy(destination, source, 5); // Copia apenas os primeiros 5 caracteres
    destination[5] = '\0'; // Adiciona o caractere nulo manualmente
    printf("Destino: %s\n", destination);
    return 0;
}
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

strcat - Concatena duas strings

A função strcat adiciona a string source ao final da string destination.

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

int main() {
    char destination[50] = "Hello, ";
    char source[] = "world!";
    strcat(destination, source);
    printf("Resultado: %s\n", destination);
    return 0;
}
```



strncat - Concatena os primeiros n caracteres de uma string a outra

A função strncat adiciona os primeiros n caracteres da string source ao final da string destination

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

int main() {
    char destination[50] = "Hello, ";
    char source[] = "world!";
    strncat(destination, source, 3); // Adiciona apenas os primeiros 3 caracteres
    printf("Resultado: %s\n", destination);
    return 0;
}
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

strcmp - Compara duas strings

A função strcmp compara duas strings lexicograficamente.

Retorna:

- 0 se as strings são iguais.
- Um valor negativo se a primeira string for menor que a segunda.
- Um valor positivo se a primeira string for maior que a segunda.

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

int main() {
    char str1[] = "Hello";
    char str2[] = "World";
    int result = strcmp(str1, str2);
    if (result == 0) {
        printf("As strings são iguais\n");
    } else if (result < 0) {
        printf("str1 é menor que str2\n");
    } else {
        printf("str1 é maior que str2\n");
    }
    return 0;
}
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

strchr - Localiza a primeira ocorrência de um caractere em uma string

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

int main() {
    char str[] = "Hello, world!";
    char ch = 'o';
    char *ptr = strchr(str, ch);
    if (ptr != NULL) {
        printf("Primeira ocorrência de '%c' encontrada em: %s\n", ch, ptr);
    } else {
        printf("Caractere '%c' não encontrado\n", ch);
    }
    return 0;
}
```



strstr - Localiza a primeira ocorrência de uma substring em uma string

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

int main() {
    char haystack[] = "Hello, world!";
    char needle[] = "world";
    char *ptr = strstr(haystack, needle);
    if (ptr != NULL) {
        printf("Substring '%s' encontrada em: %s\n", needle, ptr);
    } else {
        printf("Substring '%s' não encontrada\n", needle);
    }
    return 0;
}
```

Exercício

Escreva um programa em C que faça o seguinte:

1. Leia duas strings do usuário.
2. Compare as duas strings e informe se são iguais ou diferentes.
3. Concatene as duas strings e exiba o resultado.
4. Peça ao usuário um caractere específico e conte quantas vezes esse caractere aparece na string concatenada.
5. Exiba o comprimento da string concatenada.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Escreva um programa em C que faça o seguinte:

1. Leia uma string do usuário.
2. Verifique se a string é um palíndromo.
3. Informe ao usuário se a string é ou não um palíndromo.

Um palíndromo é uma palavra que escrita de trás para frente continua igual. Exemplo: ana, radar, rotor



JESUÍTAS BRASIL



Somos infinitas possibilidades

Arquivos

Para usar as funções de manipulação de arquivo o programador deve incluir a **STDIO.H**
Para usar um arquivo em C é necessário abrí-lo. Para tanto, a linguagem C possui o comando **fopen**.

```
arq = fopen("ArqGrav.txt", "rt");
```

O primeiro parâmetro é o nome do arquivo, o segundo a forma de abertura:

"wt": abertura para gravação, arquivo texto

"rt": abertura para leitura, arquivo texto

A função fopen retorna um "apontador" para o arquivo caso consiga abrí-lo, caso contrário, retorna a constate **NULL**.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Arquivos

```
FILE *arq;  
int result;  
char Str[50];  
  
arq = fopen("ArqGrav.txt", "rt");  
  
if (arq == NULL)  
{  
    printf("Problemas na CRIACAO do arquivo\n");  
    return;  
}
```



Arquivos

Para se trabalhar com Arquivos devemos ter sempre em mente:

- Se o arquivo não existe, devo criar o arquivo.
 - Uma vez criado, este arquivo está aberto e pronto para ser preenchido por dados.
- Se o arquivo já existe,
 - Então devo abrir este arquivo para ler os dados e imprimir ou ler e adicionar novos dados.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Arquivos Texto

Leitura

Para leitura de arquivos texto pode-se usar a função **fgets** ou **fscanf**.

Leitura com FGETS

A função FGETS lê uma linha inteira de uma vez.

```
Exemplo: result = fgets(Linha, 100, arq);  
          // o 'fgets' lê até 99 caracteres ou até o '\n'
```

Se a função for executada com sucesso, **fgets** retorna o endereço da string lida, caso contrário retorna NULL.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Arquivos Texto

Leitura

Para leitura de arquivos texto pode-se usar a função **fgets** ou **fscanf**.

```
FILE *file;
char line[256];

// Abre o arquivo em modo de leitura
file = fopen("arquivo.txt", "r");
// Lê cada linha do arquivo usando fgets
while (fgets(line, sizeof(line), file)) {
    // Remove o caractere de nova linha, se presente
    line[strcspn(line, "\n")] = 0;
    printf("%s\n", line);
}

// Fecha o arquivo
fclose(file);
```

Leitura com FSCANF

A função **FSCANF** funciona como a função SCANF, porém, ao invés de ler os dados de teclado, estes dados são lidos de arquivo.

Exemplo:

```
int i, result;  
float x;  
result = fscanf(arq, "%d%f", &i, &x);
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

Leitura com FSCANF

```
// Lê cada linha do arquivo usando fscanf
while (fscanf(file, "%49s %d", nome, &idade) == 2) {
    // Imprime os dados lidos
    printf("Nome: %s, Idade: %d\n", nome, idade);
}

// Fecha o arquivo
fclose(file);
```



Sigla	Significado
"r"	Abre um arquivo para leitura em modo texto. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"w"	Cria um arquivo em modo texto para gravação. Se o arquivo já existir, os dados serão apagados e novos dados serão incluídos.
"a"	Abre um arquivo em modo texto para adição de novos dados, a partir do seu final. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"r+"	Abre um arquivo em modo texto para atualização, ou seja, tanto para leitura como para gravação. Se o arquivo não existir ou der falha na abertura é retornado NULL.
"w+"	Cria um arquivo em modo texto para atualização, ou seja, tanto para leitura como para gravação. Se o arquivo já existir o conteúdo anterior será destruído.
"a+"	Abre um arquivo em modo texto para atualização, gravando dados novos a partir do seu final. Se o arquivo não existir ele será criado.



Gravação

Para gravação de arquivos texto usa-se as funções fputs e fprintf.

Exemplo de fputs:

```
result = fputs(Str, arq);
```

Se a função NÃO for executada com sucesso, **fputs** retorna a constante **EOF**.



JESUÍTAS BRASIL



UNISINOS

Somos infinitas possibilidades

```
char Str[100];
FILE *arq;

arq = fopen("ArqGrav.txt", "wt"); // Cria um arquivo texto para gravação
if (arq == NULL) // Se não conseguiu criar
{
    printf("Problemas na CRIACAO do arquivo\n");
    return;
}
strcpy(Str, "Linha de teste");
result = fputs(Str, arq);
if (result == EOF)
    printf("Erro na Gravacao\n");
fclose(arq);
```



FPRINTF

```
result = fprintf(arq, "Linha %d\n", i);
```

Se a função **fprintf** for executada com sucesso, devolve o número de caracteres gravados.

Se a função NÃO for executada com sucesso, retorna a constante **EOF**.



JESUÍTAS BRASIL



Somos infinitas possibilidades

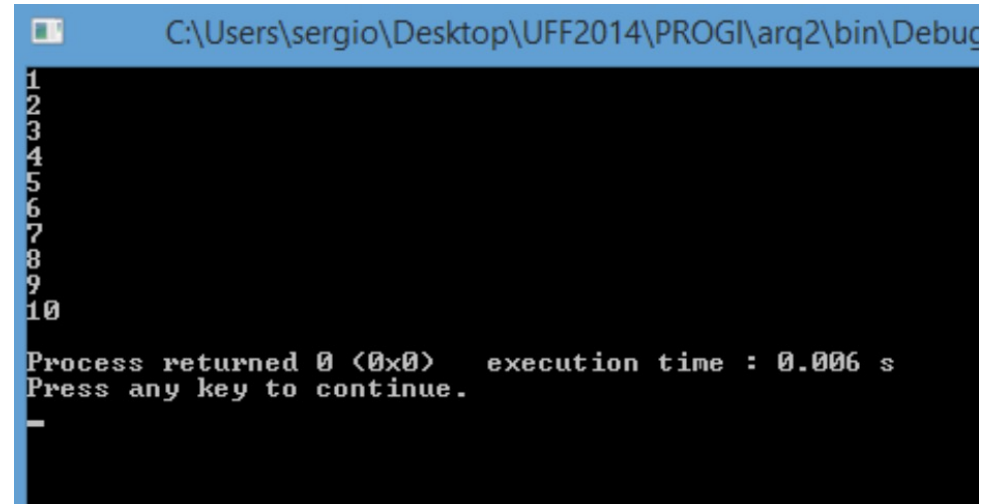
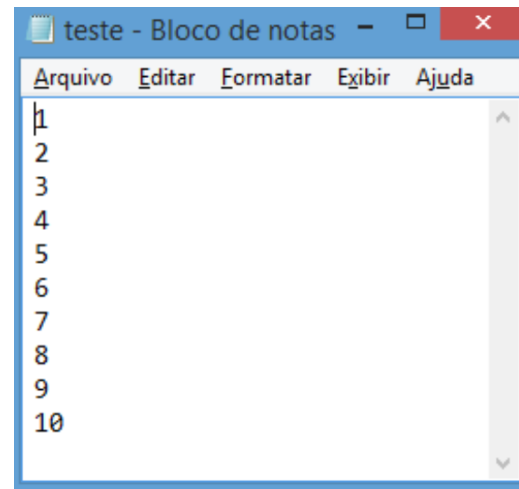

```
arq = fopen("ArqGrav.txt", "wt"); // Cria um arquivo texto para gravação
if (arq == NULL){
    printf("Problemas na CRIACAO do arquivo\n");
    return;
}

for (i = 0; i<10;i++){
    // A funcao 'fprintf' devolve o número de bytes gravados
    // ou EOF se houve erro na gravação
    result = fprintf(arq,"Linha %d\n",i);
    if (result == EOF)
        printf("Erro na Gravacao\n");
}

fclose(arq);
```



```
#include <stdio.h>
int main()
{
    FILE *f = fopen("teste.txt", "r");
    int i;
    while (fscanf(f, "%d", &i) == 1){
        printf("%d\n", i);
    }
    fclose(f);
}
```



JESUÍTAS BRASIL

UNISINOS

Somos infinitas possibilidades

```
char url[]="arquivo1.txt",
nome[20];
float nota1, nota2, nota3;
FILE *arq;
arq = fopen(url, "r");
if(arq == NULL){
    printf("Erro, nao foi possivel abrir o arquivo\n");
}
else{
    while( (fscanf(arq,"%s %f %f %f\n", nome, &nota1, &nota2, &nota3))!=EOF ){
        printf("%s teve media %.2f\n", nome, (nota1+nota2+nota3)/3);
    }

    fclose(arq);
}
return 0;
```



Escreva um programa em C que peça 3 notas de um aluno (Matemática, Física e Química), e salve esses dados em um arquivo chamado "notas.txt", que deve ter, ao final, a média das três disciplinas.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Crie um programa em C que faça o seguinte:

1. Crie um arquivo de texto chamado pessoas.txt.
2. O programa deve permitir ao usuário inserir informações sobre várias pessoas (nome, idade e altura).
3. Grave essas informações no arquivo pessoas.txt.
4. Leia as informações do arquivo e exiba-as no console.



JESUÍTAS BRASIL



Somos infinitas possibilidades

Lendo arquivos binários

Arquivos binários têm seu conteúdo armazenado no formato binário e, por isso, ocupam menos bytes que os arquivos de texto. Além disso, não precisam de separadores entre os campos e os registros. Para trabalhar com arquivos binários também precisamos da biblioteca `stdlib.h`



JESUÍTAS BRASIL



Somos infinitas possibilidades

Abrindo arquivos binários

Abrindo um arquivo binário: o parâmetro rb significa leitura de arquivo binário. Para escrever podemos usar wb

```
FILE *fileInt = fopen("inteiros.bin", "rb");  
if (fileInt == NULL) {  
    return 1;  
}
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

Escrevendo arquivos binários

O comando `fwrite` recebe um ponteiro para a variável que tem os dados o tamanho do dado a ser escrito, o número de elementos e o arquivo.

```
int fwrite (void *pt, int tam, int num, FILE *arq)
```

```
int numeros[5] = {1, 2, 3, 4, 5};
```

```
fwrite(numeros, sizeof(int), 5, file);  
fclose(file);
```



Lendo arquivos binários

O comando `fread` recebe um ponteiro para a variável que tem os dados o tamanho do dado a ser escrito, o número de elementos e o arquivo.

```
int fread(void *pt, int tam, int num, FILE *arq)
```

```
int numerosInt[5];  
fread(numerosInt, sizeof(int), 5, file);  
fclose(file);
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

Verificando se chegou ao fim de um arquivo

Podemos testar se chegamos ao final de um arquivo com o comando `feof`

```
int feof(FILE *arq)
```

```
int numero;
while (!feof(file)) {
    if (fread(&numero, sizeof(int), 1, file) == 1) {
        printf("Numero: %d\n", numero);
    }
}
```



JESUÍTAS BRASIL



Somos infinitas possibilidades

Escreva um programa o qual grava um vetor de 5 inteiros em um arquivo binário. Logo após, em outro programa C, escreva uma função para ler este arquivo e imprimir na tela.

Crie um programa em C que faça o seguinte:

1. Crie um arquivo de texto chamado strings.txt contendo várias strings, uma por linha.
2. O programa deve ler as strings do arquivo strings.txt.
3. Verifique se cada string é um palíndromo.
4. Escreva os resultados em um arquivo chamado resultados.txt, indicando se cada string é ou não um palíndromo.



JESUÍTAS BRASIL



Somos infinitas possibilidades