

- Entender o conceito de string
- Criar e Inicializar strings
- Acessar os elementos de uma string
- Ler e Escrever uma string
- Utilizar as funções básica de manipulação de strings



FUNDAÇÃO CENTRO DE ANÁLISE. PESOUISA
E INOVAÇÃO TECNOLÓGICA

Cadeia de Caracteres

- Strings são sequências ou conjuntos de caracteres diversos, sendo representadas entre aspas duplas.
- Em programação, cada símbolo armazenado na memória é indexado por um valor numérico;
- Esse tipo de literal armazena dados que contêm letras, dígitos e/ou símbolos especiais.
- Na linguagem C, não existe um tipo string explícito, isto quer dizer que não existe uma palavra reservada que declara uma variável para armazenar uma string ou cadeia de caracteres.



Cadeia de Caracteres

Exemplo: constante string

- "Maria" string de comprimento 5

- "123" string de comprimento 3

- "0" string de comprimento 1

– "A" string de comprimento 1

Então para armazenar o nome de uma pessoa, por exemplo, é necessária a criação de um vetor do tipo char, que irá possuir uma posição para cada letra ou dígito.



Cadeia de Caracteres

Suponha que na entrada de dados foi digitado: ALGORITMOS

char palavra[14];
scanf("%s", palavra);

Memória Principal (MP)

Palavra

A L G O R I T M O S 10

0 1 2 3 4 5 6 7 8 9 10 11 12 13

Posições dos caracteres dentro da variável

Em algumas linguagens, a numeração poderá começar com 1

- Os caracteres são um tipo de dado: char
- O C trata os caracteres ('a', 'b', 'x', etc) como sendo variáveis de 1 *byte* (8 bits)
- Um char também armazena número inteiro
- Este número é conhecido como o código ASCII correspondente ao caractere



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA.

Caracteres

```
#include <stdio.h>
#include <stdio.h>
                                    #include <conio.h>
#include <conio.h>
                                    int main()
int main()
                                                       Imprime o
                                                     caractere como
                                                      inteiro (68)
                                       char ch;
   char ch;
                                       ch='D';
   ch='D';
                                       printf ("%d",ch);
   printf ("%c",ch);
                                       getch();
   getch();
   return(0);
                                       return(0);
                Imprime o caractere
                  informado (D)
```

Funções para leitura

- Para terminal DOS e Windows usa-se getch() e getche()
- Ambas retornam o caractere pressionado
- getch() apenas retorna o caractere pressionado sem imprimi-lo na tela
- getche() imprime o caractere na tela antes de retorná-lo
- Ambas estão na biblioteca <conio.h>

Funções para leitura

- Geralmente, estas funções anteriores não são disponíveis em ambiente UNIX (compiladores cc e gcc), pois não são do padrão ANSI
- Por isso se recomenda o uso do getchar() da biblioteca padrão ANSI <stdio.h>



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA E INOVAÇÃO TECNOLÓGICA Caracteres

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main ()
                                             Imprime o
                                             caractere
                                             informado
   char ch;
   ch=getchar();
   printf ("Voce pressionou a tecla %c",ch);
   getch();
   return(0);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA Caracteres

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main ()
                                             Imprime o
                                             caractere
                                             informado
   char ch;
   scanf("%c", &ch);
   printf ("Voce pressionou a tecla %c",ch)
   getch();
   return(0);
```

- String é um nome que usamos para definir uma sequência de caracteres adjacentes na memória do computador.
- Essa sequência de caracteres, pode ser uma palavra ou uma frase, que é armazenada na forma de um array do tipo char.
- Por ser a string um array de caracteres, sua declaração segue as mesmas regras da declaração de um array convencional.

char str[6]; tipo_dados_char nome_vetor[tamanho]

FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA

Estrutura de Dados e Algoritmos I

Strings

- Não existe um tipo String em C.
- Strings em C são vetores do tipo char que terminam com '\0'.
- Para literais string, o próprio compilador coloca '\0'.

```
#include <stdio.h>
#include <stdlib.h>
int main(){
     char re[8] = "lagarto"; //tamanho máximo de 7 caracteres
     printf ("%s", re);
     system("pause");
     return(0);
}
```

Inicialização de Strings

- Para declarar uma string, usa-se o seguinte formato
 - char nome_da_string [tamanho];
- Para inicializar uma string, pode-se usar o mesmo princípio definido na inicialização de vetores e matrizes.

```
- char str[10] = \{'J', 'o', 'a', 'o', '\setminus 0'\};
```

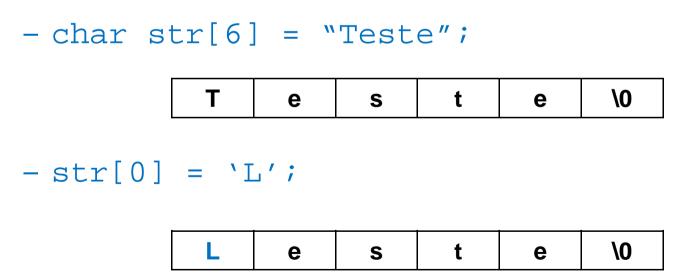
 Percebe-se que essa forma n\u00e3o \u00e9 muito pr\u00e1tica. Por isso, tamb\u00e9m pode ser feita por meio de aspas duplas.

```
- char str[10] = "Joao";

Possui a vantagem de já inserir o caractere \\0'
```

Acessando um elemento da String

 Outro ponto importante na manipulação de strings é que, por se tratar de um array, cada caractere pode ser acessado individualmente por indexação.



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA.

Estrutura de Dados e Algoritmos I

Strings

```
#include <stdio.h>
#include <stdlib.h>
#include <conio.h>
int main(){
   char str[6] = "Teste";
   printf ("Palavra: %s",str);
   str[0] = 'L';
   printf ("\n Nova Palavra: %s",str);
   getch();
   return(0);
```

- Se quisermos ler uma string fornecida pelo usuário podemos usar a função gets() em vez do scanf()
- Na linguagem C, o índice começa em zero;
- A primeira letra da string sempre estará na posição 0



Quando usamos a função scanf() para ler uma string, o símbolo de & antes do nome da variável não é utilizado.



Para ler uma String

Comando scanf vs. gets

```
A função scanf() lê apenas strings digitadas sem espaços, ou seja, palavras.
```

```
#include <stdio.h>
#include <stdio h>
                                       #include <stdlib.h>
#include <stdlib.h>
                                                             Lê com espaços em
                                       int main(){
int main(){
                                                            brancos até encontrar
                                          char re[8];
                                                              uma tecla enter.
  char re[8];
                                           printf ("Digite o seu nome: ");
  printf ("Digite o seu nome: ");
                                          gets(re);
  scanf("%s", re);
                                          printf ("Oi %s \n", re);
  printf ("Oi %s \n", re);
                                          system("pause");
  system("pause");
                                          return (0);
  return (0);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA

Estrutura de Dados e Algoritmos I

Strings

```
#include <stdio.h>
                       Tamanho máximo da
#include <stdlib.h>
                       String 99 caracteres
int main (){
   char string[100];
   printf("Digite uma string: ");
   gets(string);
   printf("\n\nVoce digitou %s",string);
   system("pause");
   return(0);
```

Recomendação:

- Use a sintaxe de fgets(STRING, TAMANHO, STREAM)
 - STRING variável onde a String será armazenada
 - TAMANHO é o tamanho máximo da string
 - STREAM é de onde os caracteres serão lidos, para ler do teclado o valor padrão para isto é stdin

```
printf("Digite uma frase qualquer:");
fgets(frase,50,stdin);
printf("\n");
```

FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA E INOVAÇÃO TECNOLÓGICA **Strings**

```
#include <stdio.h>
                      Tamanho máximo da
#include <stdlib.h>
                       String 99 caracteres
int main (){
   char string[100];
   printf("Digite uma string: ");
   fgets(string, 10, stdin);
   printf("\n\nVoce digitou %s",string);
   return(0);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA **Strings**

```
#include <stdio.h>
#include <stdlib.h>
int main(){
   char str[10] = "Joao";
   printf("\n\nString: %s", str);
   printf("\nSegunda letra: %c", str[1]);
   str[1] = 'U';
   printf("\nAgora a terceira letra eh: %c", str[2]);
   printf("\n\nString resultante: %s \n", str);
   system("pause");
   return(0);
```

- Funções que manipulam strings trabalham usualmente com a referência para o início da sequência de caracteres, ou seja, com um apontador para a string
- Por conta disso!
 - não é possível copiar o conteúdo de duas strings s2 para s1 simplesmente por atribuição

NÃO faça isso!

Exemplo:

```
s1 = s2; /* copia o endereco! */
if (s1 != s2) /* compara os enderecos! */
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA

Estrutura de Dados e Algoritmos I

Funções de Manipulação de Strings

- Abaixo as principais funções usadas para a manipulação de strings em linguagem C.
- Essas funções estão no arquivo cabeçalho string.h

• strcpy	• strupr
• strcat	• strlwr
• strlen	• strrev
• strcmp	• strset



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA. E INOVAÇÃO TECNOLÓGICA Função strcpy()

A linguagem C não suporta a atribuição de um array para outro.

str **⇒** João"

A função strcpy() copia a string-origem para a string- destino

Uso:

```
strcpy(string_destino, string_origem);
```



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA E INOVAÇÃO TECNOLÓGICA Função strcpy()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char str1[100],str2[100],str3[100];
   printf ("Entre com uma string: ");
   gets (str1);
   strcpy (str2,str1);
   strcpy (str3,"Voce digitou a string: ");
   printf ("\n\n\%s %s \n", str3, str2);
system("pause");
return 0;
```

Função strcat()

A *string* de origem permanecerá inalterada e será anexada ao fim da *string* de destino.

Basicamente, essa operação consiste em copiar uma string para o final de outra string.

```
Uso:
```

```
strcat (string destino, string origem);
```



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA E INOVAÇÃO TECNOLÓGICA Função strcat()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char str1[100],str2[100];
   printf ("Entre com seu nome: ");
   qets (str1);
   strcpy (str2, "Bom dia, ");
   strcat (str2,str1);
   printf ("\n\n%s\n",str2);
system("pause");
return 0;
```



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA E INOVAÇÃO TECNOLÓGICA Para concatenar duas strings

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>
int main(){
  char str1[40],str2[10],str3[40]="teste";
  strcpy(str1,"inicio");
  strcpy(str2,"FIM");
  strcat(str1,str2);
  strcat(str3,str1);
  printf("%s\n",str1);
  printf("%s\n",str3);
  system("pause");
  return (0);
```

Função strlen()

A função strlen() retorna o comprimento da string fornecida;

O terminador nulo não é contado;

```
Uso:
```

```
strlen(string);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA. E INOVAÇÃO TECNOLÓGICA Função strlen()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
  int tam;
  char str[100];
  printf("Entre com uma frase: ");
  gets(str);
  tam=strlen(str);
printf("\n\nA frase que você digitou tem %d caracteres \n",tam);
system("pause");
return 0;
```

A função strcmp() compara a string 1 com a string 2;

Se as duas forem idênticas a função retorna zero;

Se elas forem diferentes a função retorna:

1 se a string2 for maior

-1 se a string2 for menor

USO:

```
strcmp(string1,string2);
```



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA.

Função stremp()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
  char str1[100],str2[100];
  printf ("Entre com uma string: ");
                                           qets(str1);
  printf ("\n\nEntre com outra string: "); gets (str2);
  if (strcmp(str1,str2)==0)
    printf ("\n\nAs duas strings são iguais.\n");
  else
    printf("\n\nAs duas strings s\( \tilde{a} \) diferentes.\n");
system("pause");
return(0);
```



FUNDAÇÃO CENTRO DE ANÁLISE. PESQUISA. E INOVAÇÃO TECNOLÓGICA

Para comparar duas strings

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
  char re[80];
  printf ("Digite a senha: ");
  gets(re);
  if (strcmp(re,"laranja")==0) {
     printf ("Senha correta\n");
  else {
     printf ("Senha invalida\n");
  system("pause");
  return (0);
```

Função strupr()

Função strupr() converte uma string para maiúscula

```
USO:
    strupr(string);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA Função strupr()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char string[20];
   printf("Entre com a string :");
   scanf("%s",&string);
   printf("Convertendo : %s\n",strupr(string));
system("pause");
return(0);
```

Converte uma string para minúscula

```
USO:
    strlwr(string);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA Função strlwr()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char string[20];
   printf("Entre com a string :");
   scanf("%s",&string);
   printf("Convertendo : %s\n",strlwr(string));
system("pause");
return(0);
```



Função strrev()

Função strrev();

Inverte uma string dada

USO:

strrev(string);



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA Função strrev()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char string[20];
   printf("Entre com a string :");
   scanf("%s",&string);
   printf("Invertendo : %s\n",strrev(string));
system("pause");
return(0);
```

Função strset ()

Substitui todos os caracteres da *string* pelo mesmo caractere usado na função strset()

```
USO:
    strset(string,char);
```



FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA E INOVAÇÃO TECNOLÓGICA Função strset ()

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
   char string[20];
   printf("Entre com a string :");
   scanf("%s",&string);
   printf("Substituindo: %s\n",strset(string, '*'));
system("pause");
return(0);
```



Convertendo string em número

 Estas funções fazem parte do arquivo de cabeçalho stdlib.h

FUNÇÃO	CONVERTENDO
atof(string)	float
atoi(string)	int
atol(string)	long int
strtod(string)	double
strtol(string)	long



Convertendo string em número

```
#include<stdin.h>
#include<stdlib.h>
int main()
  char str[80];
  int resultado;
  float resultado2;
  scanf("%s", &str);
  resultado = atoi(str);
  printf("Conversão:%s-%d\n", str, resultado);
  scanf ("%s", &str);
  resultado2 = atof(str);
  printf("Conversão:%s-%f\n", str, resultado2);
  system("pause");
  return 0:
```



Estrutura de Dados e Algoritmos I Exemplo de Uso das Funções FUNDAÇÃO CENTRO DE ANÁLISE, PESQUISA

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char palavra[30], palavra2[30], teste[30];
    int tam, comp;
    printf ("\n Informe uma string: ");
    gets (palavra);
    printf ("A palavra digitada eh %s\n", palavra);
    printf ("\n Informe uma segunda string: ");
    gets (palavra2);
    printf ("A palavra digitada eh %s\n", palavra2);
     strcpy (teste, "aula teste");
    printf ("A palavra copiada eh %s\n", teste);
```



Estrutura de Dados e Algoritmos I Exemplo de Uso das Funções

```
strcat (teste, "1");
  printf ("A palavra concatenada eh %s\n", teste);
  tam=strlen(palavra);
  printf ("\nO tamanho da primeira string eh %d", tam);
  comp=strcmp (palavra, palavra2);
  if (comp==0)
    printf ("\nSao iguais: %d", comp);
        else
            printf ("\nSao diferentes: %d", comp);
   printf ("\nMaiusculo: %s", strupr(palavra));
   printf ("\nMinusculo: %s\n", strlwr(palavra));
   system("pause");
return (0);
```



 Escreva um programa que leia um nome completo de uma pessoa e conte quantas vogais esse nome possui. Ao final, imprimir o nome completo e a quantidade de vogais:

2. Escreva um programa que receba uma string S e dois valores inteiros não negativos *i* e *j*. Em seguida, imprimir os caracteres contidos no seguimento (intervalo) que vai de *i* a *j* da string S:



Estrutura de Dados e Algoritmos I Exercício de Fixação

3. Criar um algoritmo que entre com uma palavra e imprima conforme o exemplo a seguir:

Palavra: PARALELEPIPEDO

P

A

R

A

L

E

L

P

l P

E

0

