

Cadenas de texto

Cadenas de texto en lenguaje C

Para trabajar con cadenas de texto en el lenguaje C, es necesario invocar la librería al comienzo del código con la directiva “include”:

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

Una cadena de texto o “string” está conformada por varios caracteres contiguos, por lo tanto, para utilizar una cadena de texto se declara un vector del tipo de dato CHAR que representa a varios caracteres contiguos (uno al lado del otro).

Por ejemplo:

```
char cadena[100];
```

Observación: la longitud de la variable “cadena” es de 99 caracteres, al introducir C el carácter de fin de cadena ‘\0’ da un total de 100 caracteres. La variable “cadena” puede contener una frase de hasta 99 caracteres y siempre se calcula un carácter adicional que representará al “fin” o terminación de la cadena.

Asignación de valores a una cadena de texto

La asignación de valores a una variable en el lenguaje C es por medio del signo “=”. En el caso de una cadena de texto esta forma de asignación no se utiliza, puede provocar un error.

Existen varios métodos para asignar valores a una cadena de texto:

1. Por teclado, usando el comando:

```
scanf("%s", cadena);
```

2. Con el comando “STRING COPY”:

```
strcpy(cadena, "valor de la cadena");
```

3. Con el comando “SPRINTF”:

```
sprintf(cadena, "valor de la cadena");
```

Observación: el comando “SPRINTF” no necesita la librería “string.h”.

Copiar valores desde una cadena de texto hacia otra cadena

- Con el comando “STRING COPY”, desde “cadena2” hacia “cadena”:

```
strcpy(cadena,cadena2);
```

- Con el comando “SPRINTF” desde “cadena2” hacia “cadena”:

```
sprintf(cadena, "%s",cadena2);
```

Observación: el comando “SPRINTF” no necesita la librería “string.h”.

Copiar valores desde un valor entero hacia una cadena de texto

- Con el comando “SPRINTF” desde un entero hacia una “cadena”:

```
sprintf(cadena, "%d",numero);
```

Observación: el comando “SPRINTF” no necesita la librería “string.h”.

Copiar valores desde un valor *float* hacia una cadena de texto

- Con el comando “SPRINTF” desde un *float* hacia una “cadena”:

```
sprintf(cadena, "%f",decimal);
```

Observación: el comando “SPRINTF” no necesita la librería “string.h”.

Copiar valores desde una cadena de texto hacia un entero

- Con el comando “atoi” desde una “cadena” hacia un entero:

```
int numero = atoi(cadena); //cadena tiene el valor "10".
```

Comparar cadenas de texto

- Con el comando “STRING COMPARE”:

```
//La comparación distingue entre mayúsculas y minúsculas
//El lenguaje es case sensitive.
if(strcmp(cadena1,cadena2)==0)
{
    //son iguales.
}else{
    //son distintas.
}
```

Concatenación de cadena de texto

- Con el comando "STRING CAT":

```
strcat(cadena1, cadena2);
```

- Con el comando "PRINTF":

```
//Concatenación de cadena.  
sprintf(cadena1, "%s %s", cadena1, cadena2);  
  
//Concatenación de cadena con valores enteros.  
sprintf(cadena1, "%s %d", cadena1, numero);  
  
//Concatenación de enteros.  
sprintf(cadena1, "%d %d", num1, num2);
```

Observación: el carácter de separación utilizado en el ejemplo de código es un espacio, se pueden utilizar otros, ejemplo '|' o ';'.

Cantidad de caracteres de una cadena de texto

- Con el comando "STRING LEN":

```
strlen(cadena);
```

- Con el comando "SIZEOF":

```
sizeof(cadena);
```

Ejemplo de iteración con cadenas de texto

```
#include <stdio.h>  
#include <stdlib.h>  
  
int main()  
{  
    int i = 0;  
    int j = 0;  
    char cadena[] = "Ejemplo de cadena";  
  
    while (cadena[i] != '\0')  
    {  
        printf("%c", cadena[i]);  
        i = i + 1;  
    }  
}
```

```

printf("%cUsando \\0: ", '\n');

while (cadena[j] != '\0')
{
    printf("Iteracion %d - ", j + 1);
    j = j + 1;
}

printf("%cUsando sizeof: ", '\n');

j = 0; //Reinicializar j

while (j < sizeof(cadena))
{
    printf("Iteracion %d - ", j + 1);
    j = j + 1;
}

printf("\nLongitud de la cadena es %d", sizeof(cadena));

return 0;
}

```