

Strings (Cadenas)

Curso de Estructura de datos y algoritmos 1 – Universidad ORT Uruguay

Conceptos generales

1. ¿Que son las strings?

En C no existe un tipo nativo para manipular cadenas de caracteres o más conocidos en el mundo de la programación como strings. Sin embargo, existe una biblioteca llamada “string.h” para su manipulación.

Una cadena en C es un array de caracteres de una dimensión que debe terminar con el caracter especial \0 (contrabarra cero), llamado carácter de terminación.

El formato para declarar y reservar memoria para una cadena utilizando memoria dinámica como se comentó en la lectura anterior es:

```
char* cadena = new char[n];
```

donde n es un entero positivo ≥ 1 y representa la longitud real de la cadena que se quiera representar.

Un ejemplo de la declaración de una cadena es:

```
char* cadena = new char[6];
```

Dado a que la representación interna de una cadena de caracteres es terminada por el carácter \0, para un texto de largo n, debemos reservar un vector de largo n+1.

Podemos utilizar varias formas de inicialización de los vectores de caracteres, al igual que lo podemos hacer con cualquier otro tipo de vector.

```
char* cadena = new char[6]{'H','e','l','l','o','\0'};  
char* cadena = new char[]{'H','e','l','l','o','\0'};
```

Para acceder a un elemento de la cadena puede hacerse de la misma forma que el acceso a un elemento de un vector, en definitiva no deja de ser un vector al igual que cualquier.

A:	'H'	'e'	'l'	'l'	'o'	'\0'
	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]

```
A[i];
```

Siendo i un entero positivo, y n el largo del vector se cumple que para todo i $0 \leq i < n$.

Si bien C nos permite acceder a las posiciones mayores a n, dichas posiciones de memoria no pertenecen al array que se quiere recorrer.

2. Ejemplo introductorio

```
void main(){
    char* cadena = new char[3]{'O','K','\0'};
    int largoCadena = largo(cadena);
    cout << largoCadena << endl;

}

// PRE: -
// POS: Retorna el largo de la cadena str
int largo(char* str) {
    unsigned int largo = 0;
    while(*str != '\0') {
        largo++;
        str++;
    }
    return largo;
}
```

Como se puede notar en el ejemplo de arriba tenemos definida una función llamada `largo` que nos permite calcular el largo de una cadena, utilizando una variable entera para acumular el largo, itera mientras que el puntero no apunte al carácter de terminación `'\0'`. En cada iteración se avanza en el puntero, y se suma 1 al contador del largo de la cadena.

Es importante notar que esta función nos va a retornar el largo real de la cadena, es decir si nuestra cadena representa la palabra “Hola”, retorna 4.

Esta función al igual que muchas otras se encuentran definidas en la biblioteca de C mencionada anteriormente llamada “string.h”.

Para utilizarlas debemos incluir la biblioteca a nuestro proyecto:

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

A continuación se irán a mencionar algunas de las funciones más importantes:

- `strlen(<cadena>)`: Devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.
- `strcpy(<cadena_destino>, <cadena_origen>)`: Copia el contenido de `<cadena_origen>` en `<cadena_destino>`.
- `strcat(<cadena_destino>, <cadena_origen>)`: Concatena el contenido de `<cadena_origen>` al final de `<cadena_destino>`.
- `strcmp(<cadena1>, <cadena2>)`: Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales, un número negativo si `<cadena1>` es menor que (precede alfabéticamente a) `<cadena2>` y un número positivo (mayor que cero) si `<cadena1>` es mayor que `<cadena2>`.

Ejercicios

1. Dada la siguiente función:

```
int misterio1(char* str){
    char* aux = str;
    while (*aux != '\0'){
        aux = aux + 1;
    }
    return aux - str;
}
```

- 1) ¿Que realiza la misma?
- 2) ¿Como se llama esta función en la biblioteca string.h?

2. Dada la siguiente función:

```
void misterio2(char* str1, char* str2){
    while (*str2 != '\0') {
        *str1 = *str2;
        str1++;
        str2++;
    }
    *str1 = '\0';
}
```

```
void misterio3(char* str1, char* str2){
    while (*str1++ = *str2++)
}
```

- 1) ¿Que realizan cada una de las funciones anteriores?
- 2) ¿Como se llaman estas funciones en la biblioteca string.h?

3. Dada la siguiente función:

```
int misterio4(char* str1, char* str2) {
    for (; *str1 == *str2; str1++, str2++) {
        if (*str1 == '\0'){
            return 0;
        }
    }
    return *str1 - *str2;
}
```

- 1) ¿Que realiza la misma?
- 2) ¿Como se llama esta funcion en la biblioteca string.h?