Módulo 6: Tópicos Avanzados

3. Strings

Strings terminados en nulo.

C++ soporta 2 tipos de strings. El primero de ellos son los strings terminados en nulo; el otro tipo es utilizando la clase String, que te da una forma orientada a objetos para maejar los strings. En esta sección nos enfocaremos a los strings terminados en nulo.

Nota. Dependiendo de la implementación del compilador, la biblioteca estándar de C++ podría llamarse string.h, o bien, cstring.h.

Declarar un string

Un string se representa internamente como un arreglo de caracteres; por lo tanto se declara de esta forma con el siguiente formato:

char nomVar[tamaño+1]

C++ agrega un caracter nulo (\0) al final de un string para indicar en dónde termina. Esa es la razón por la que se debe declarar reservando un caracter más de lo que se requiere guardar.

Por ejemplo, podemos declarar el string Palabra que contenga espacio para guardar un string de hasta 15 caracteres de la siguiente forma:

char Palabra[16];

y se puede utilizar un valor string para inicializar el string en su declaración de la siguiente forma:

char Palabra[15] = "Clase";

este string internamente se representa así:

Palabra

C	l	a	s	e	\0									
0	1	2	3	4	5	6	7	8	9	10	11	12	13	14

Es decir, se coloca un caracter en cada una de las casillas del arreglo, y después del último caracter se agrega el caracter nulo (nota que el segundo renglón de esta tabla contiene el número correspondiente a la casilla del arreglo).

Cuando usas un valor string en el programa, es decir una lista de caracteres encerrados entre comillas, en memoria se crea un string terminado en nulo; es decir, siempre una constante string se maneja como un string terminado en nulo.

Como el string es en realidad un arreglo de caracteres, podemos accesar cada uno de los caracteres de la siguiente forma:

Nombre_string[posición]

Por ejemplo, podemos accesar el primer caracter del string Palabra utilizando:

Palabra[0]

Mostrar en la pantalla un string

Se puede mostrar un string en la pantalla utilizando cout de la forma tradicional.

Por ejemplo:

cout<<Palabra;

Leer un string

Se puede leer un string del teclado utilizando el cin de la forma tradicional. Solo es importante recordar que el cin ignora los espacios en blanco, tab o return del inicio y termina de leer cuando se encuentra con espacio en blanco, tab o return; por lo que con un cin solamente se puede leer valores String que contengan una sola palabra.

Por ejemplo:

cin>>Palabra;

También se puede leer un string del teclado utilizando getline, la diferencia con cin es que getline no ignora los espacios en blanco ni los tab y termina de leer cuando se encuentra con un Return; por lo que con getline sí es posible leer un string que contenga varias palabras.

El formato para utilizarlo es:

cin.getline(nombreString, longitud);

esta función toma del teclado cuando mucho longitud ?1 caracteres y termina de leer cuando encuentra un Return; el string leido se coloca en la variable nombreString. Este estatuto retira el caracter \n del input stream, pero no lo pone en el string.

Por ejemplo:

cin.getline(Palabra, 15);

Algunas funciones existentes en la biblioteca string.h

C++ contiene una gran variedad de funciones de manejo de strings, en esta sección se explicarán las más comunes (en algunas versiones de C++ la biblioteca podría ser cstring.h).

Función para obtener la longitud del string

Formato:

variableEntera = strlen(NombreString);

Esta función regresa la longitud de un string. El valor que regresa es el número de caracteres que

contiene el string sin contar el caracter nulo de terminación.

Por Ejemplo:

```
char Pal[10] = ?hola?;
int x = strlen(Pal);
```

En este ejemplo la variable x toma el valor 4.

Función para concatenar dos strings

Formato:

```
strcat(primerString, segundoString);
```

Esta función recibe como parámetro dos strings, y concatena el contenido del segundo string al final del primero. La longitud del string resultante es igual a la suma de las longitudes de los strings concatenados.

Por ejemplo:

```
char s1[10] = ?uno?;
char s2[10] = ?dos?;
strcat(s1, s2);
```

Al ejecutar estos estatutos la variable s1 toma el valor ?unodos?

Función para copiar un string a otro

Formato:

```
strcpy(primerString, segundoString);
```

Esta función copia el contenido de un string en otro. A diferencia de la concatenación, esta función borra el contenido del primer string y en su lugar pone el contenido del segundo.

Por ejemplo:

```
char s1[10] = ?uno?;
char s2[10] = ?dos?;
strcpy(s1, s2);
```

Al ejecutar estos estatutos la variable s1 toma el valor "dos"

No es posible asignar directamente el valor de una constante string a una variable de tipo string. Para hacerlo es necesario utilizar la función strcpy poniendo como segundo sting el valor a copiar.

Por ejemplo:

```
char Pal[10];
strcpy(Pal, ?algo?);
```

Después de ejecutar estos estatutos la variable Pal toma el valor ?algo?

Función para comparar dos strings

Formato:

varEntera = strcmp(primerString, segundoString);

Esta función realiza una comparación entre dos strings empezando por el primer caracter de cada string y continuando la comparación caracter por caracter (mientras sean iguales) hasta encontrar que dos caracteres difieran, o bien, que se alcance el final de alguno de los strings. El resultado de la comparación lo regresa de acuerdo con la siguiente tabla:

Si el primer string es	**strcmp regresará un valor**
menor que el segundo	< 0
igual que el segundo	== 0
mayor que el segundo	> 0

Por ejemplo:

```
char s1[10] = ?uno?;
char s2[10] = ?dos?;
int x = strcmp(s1, s2);
```

Al ejecutar estos estatutos la variable entera x tomará un valor mayor a 0

Ejemplo:

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

// Esta funcion verifica si un string es un palíndrome
// regresa 1 si es palíndrome y 0 si no lo es
int esPalindrome(char str[])
{
   int i,j;
   char aux[20];

   // construye un string que contiene los caracteres al revés
   for (i = strlen(str)-1, j=0; i>=0; i--, j++)
       aux[j]=str[i];
   aux[j]='\0'; // al construir el string no olvidar el nulo al final

   if (strcmp(aux, str) == 0)
      return 1;
   else
      return 0;
}

int main()
{
    char palabra[20], resp;

    do
```

```
    {
        cout<<"Teclea una palabra"<<endl;
        cin>>palabra;

        if (esPalindrome(palabra))
            cout<<"si es palindrome"<<endl;
        else
            cout<<"no es palindrome"<<endl;
        cout<<"Revisar otra palabra"<<endl;
        cin>>resp;
    } while ((resp == 's') || (resp == 'S'));
    return 0;
}
```

Ejercicio

Realiza un programa que pida al usuario una palabra y 2 letras. El programa debe sustituir todas las ocurrencias de la primera letra por la segunda letra; por ejemplo:

Si se teclea la palabra abecedario y la primera letra es a y la segunda z se obtendrá el siguiente string:

zbededzrio
ver solución

Ligas sugeridas

http://www.cplusplus.com/doc/tutorial/

http://www.cs.wustl.edu/~schmidt/C++/

Regresar

Siguiente módulo