

UNIVERSIDAD DE GUANAJUATO

DIVISIÓN DE INGENIERÍAS CAMPUS IRAPUATO-SALAMANCA

LIC. EN ING. EN SISTEMAS COMPUTACIONALES

ALGORITMOS Y ESTRUCTURAS DE DATOS

PROFESOR: DR. CARLOS HUGO GARCÍA CAPULÍN

NO. DE TAREA: 04

NOMBRE DE LA TAREA:

CADENA DE CARACTERES

ESTUDIANTE:

MANRÍQUEZ COBIÁN ROGELIO

FECHA DE ENTREGA:

11 DE SEPTIEMBRE DEL 2020



Problema

Implementar un programa en lenguaje C que reciba una frase y entregue como resultado una tabla como la siguiente:

"Saludos a todos los presentes"

PALABRAS	NO. CARACTERES
Saludos	7
a	1
todos	5
los	3
presentes	9

TOTAL, DE PALABRAS = 5

TOTAL, DE CARACTERES = 25

Solución Implementada:

```

18
19 #include <stdio.h>
20
21 int main (void){
22     char frase[200];
23     unsigned int id;
24     unsigned int contador_Palabras=0;
25     unsigned int contador_CaracteresPalabras;
26     unsigned int total_Caracteres=0;;
27
28     printf("Escriba una frase: " );
29     scanf("%[^\n]s", frase); fflush(stdin);
30
31     id = 0;
32     printf("PALABRAS      No. CARACTERES\n");
33     //Encontrar una palabra en especifico
34     while(frase[id] != '\0'){
35
36         //Empieza a brincar los espacios que hay en la frase
37         while (frase[id] == ' '){
38             id++; //Sumamos el indice para volver al ciclo
39         }
40         contador_Palabras++; //Contamos cada palabra que encontramos
41
42         //Recorrer el arreglo hasta encontrar el siguiente espacio
43         contador_CaracteresPalabras=0;
44         while (frase[id] != ' ' && frase[id] != '\0'){
45
46             printf("%c", frase[id]);
47
48             contador_CaracteresPalabras++; //Contamos cada caracter de la palabra
49             id ++; //Sumamos el indice para volver al ciclo
50         }
51         printf("\t      %u \n", contador_CaracteresPalabras);
52         total_Caracteres += contador_CaracteresPalabras;
53     }
54
55     printf("\n%s", frase);
56     printf ("El total de palabras es: %u", contador_Palabras);
57     printf ("El total de caracteres es: %u", total_Caracteres);
58
59     printf("\n\nPresiona <ENTER> para salir");
60
61     getchar ();
62     return 0;
63 }

```

Pruebas y Resultados:

A continuación, se verá la explicación de lo que se hizo en el programa de acuerdo para dar solución que se nos planteó desde un inicio.

Iniciamos creando un nuevo archivo en nuestro editor de textos Notepad ++ el cual lo guardaremos con el nombre de T04.c ("En mi caso tiene el nombre de Clase10, ya que este ejercicio se planteó en la clase).

Comenzamos escribiendo nuestro **"#include <stdio.h>"**, después de esto, escribiremos nuestra función "main" en el cual contendrá todo nuestro bloque de código que vayamos escribiendo.

El primer paso por realizar es declarar nuestras variables que utilizaremos en nuestro programa, la primera variable que se declara es un arreglo de tipo **"char"** con el nombre de **"frase"** y un tamaño de **"200"** caracteres; después de esto haremos 3 declaraciones de variable de tipo **"unsigned int"**, la primera declaración de este tipo de dato es un **"id"** el cual se definirá como el tamaño del arreglo, tenemos el **"contador_Palabras"** el cual nos ayudará a almacenar cada palabra sé que encuentre, luego, **"contador_CaracteresPalabras"**, el cual ayudará a contar cada carácter que se encuentre en una palabra, y por último **"total_Caracteres"** para saber cuántos caracteres cuenta la frase que el usuario ingresó .

Luego, imprimiremos en pantalla que el usuario empiece a escribir una frase; después nosotros utilizaremos un **"scanf"** para almacenar la oración en nuestra variable **"frase"** algo, raro que está dentro de nuestro **"scanf"** es la siguiente parte **"%[^\n]s"** el cual indica que por cada espacio que dé el usuario para escribir la oración ésta también la almacenará dentro del arreglo.

Ahora tenemos que inicializar nuestra variable de **"id"** en cero, ya que esta nos ayudará en cada iteración del ciclo **"while"**.

Escribiremos nuestro **primer** ciclo **"while"** con la siguiente condición: **"frase[id] != '\0'"**, ahora dentro de este primer **"while"**, crearemos un segundo y tercer **"while"**, en el **segundo** **"while"** tendremos la condición **"frase[id] == ' '"**, el **tercero** tiene como condición **"frase[id] != 0 ' ' && frase[id] != '\0' "**.

Lo que hace nuestro primer ciclo, es encontrar una palabra en específico, y cuando lo encuentre nuestro **"contador_Palabras"** empezará a incrementarse en uno cada que encuentre una palabra. Al segundo ciclo empezará a brincar los espacios que hay en la frase y empezará a incrementar nuestro **"id"** en uno para volver a empezar el ciclo y observar las particularidades del primer ciclo; Ahora inicializaremos nuestro **"contador_CaracterPalabras"** en cero, para que en la tercera condición empecemos a recorrer todo el arreglo hasta encontrar el siguiente espacio y empezar a imprimir en pantalla una pequeña tabla por cada palabra que se encontró en todo el arreglo, y para esto haremos que nuestro **"contador_CaracteresPalabras"** se incremente en uno por

cada palabra encontrada y de igual manera el “id” se incrementará en uno para volver a realizar los demás ciclos.

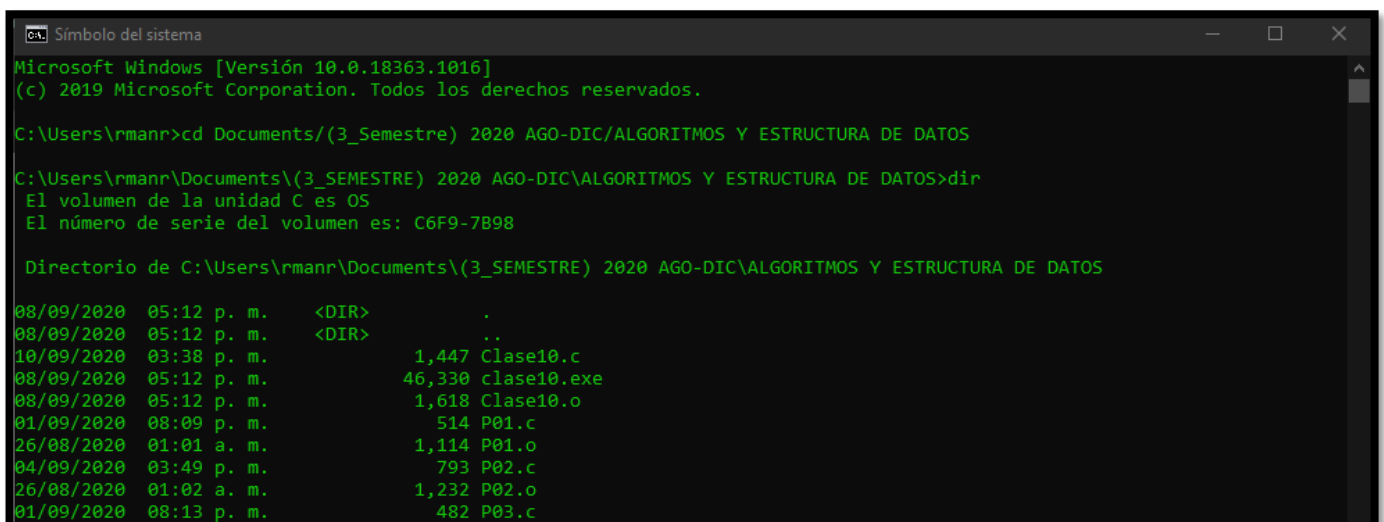
Por último, en nuestro ciclo tendremos que imprimir en pantalla los caracteres que se encontraron en cada palabra y tabularlo junto con nuestra lista de palabras encontradas; después haremos un contenedor de “**total_Caracteres** += **contador_CaracteresPalabras**” para almacenar el total de los caracteres que cuenta nuestra frase sin contar con los espacios que esta incluye.

Fuera de nuestro ciclo, solo se imprime la frase nuevamente para visualizar si esta es correcta, después de esto imprimiremos el total de palabras que se almacenó en “**contador_Palabras**” y de igual manera imprimiremos el total de caracteres que se almacenó en “**total_Caracteres**”.

Para comprobar que nuestro programa realiza lo pedido, abriremos nuestro “**CMD**” he ingresaremos hasta la carpeta donde se tiene guardado el archivo; en mi caso se encuentra en la dirección:

- *C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS*

Y con el comando <dir> veremos que el archivo se ha guardado de manera satisfactoria.

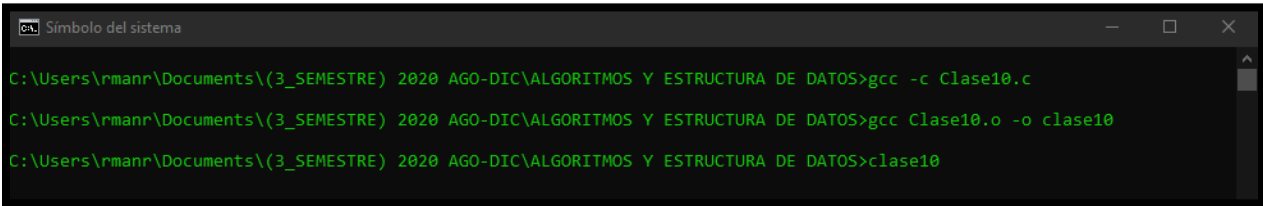


```

C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS
C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS>dir
El volumen de la unidad C es OS
El número de serie del volumen es: C6F9-7B98

Directorio de C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS
08/09/2020  05:12 p. m.    <DIR>          .
08/09/2020  05:12 p. m.    <DIR>          ..
10/09/2020  03:38 p. m.             1,447 Clase10.c
08/09/2020  05:12 p. m.      46,330 clase10.exe
08/09/2020  05:12 p. m.       1,618 Clase10.o
01/09/2020  08:09 p. m.           514 P01.c
26/08/2020  01:01 a. m.       1,114 P01.o
04/09/2020  03:49 p. m.           793 P02.c
26/08/2020  01:02 a. m.       1,232 P02.o
01/09/2020  08:13 p. m.           482 P03.c
  
```

Ahora tendremos que compilar nuestro código con los siguientes comandos que se muestran en la imagen:



```

C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS>gcc -c Clase10.c
C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS>gcc Clase10.o -o clase10
C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS>clase10
  
```

Si todo salió bien, solamente dará líneas de salto significando que todo el proceso de compilación y enlazamiento salió bien; de lo contrario, escribiste algún comando mal o tu código tiene algún error de sintaxis.

Ahora ejecutaremos nuestro programa para realizar las pruebas y observar si es lo que se quiso resolver desde un principio:



```

C:\Users\rmanr\Documents\3_SEMESTRE) 2020 AGO-DIC\ALGORITMOS Y ESTRUCTURA DE DATOS>clase10
Escriba una frase: Saludos a todos los presentes
PALABRAS      No. CARACTERES
Saludos        7
a              1
todos         5
los           3
presentes      9

Saludos a todos los presentes
El total de palabras es: 5
El total de caracteres es: 25
Presiona <ENTER> para salir
  
```

A continuación, se explicará la prueba que se realizó:

El programa nos pregunta que escribamos una frase, la frase de prueba es “**Saludos a todos los presentes**”

Después nuestro programa realizará el conteo de caracteres que contiene cada una de nuestra palabra, por ejemplo:

PALABRA	No. CARACTERES
Hola	4

Entonces el programa hará una lista que contendrá cada palabra y su tamaño de caracteres que contiene.

Después de esto, solo mostrará nuevamente la frase que se ingresó, y saldrá en pantalla el total de palabras que contiene el arreglo y el total de caracteres que contiene el arreglo sin contar los espacios que se escribieron desde un principio.

Como resultado final, tenemos la tabla que se planteó desde un principio.

Como conclusión de este reporte es que es muy interesante todo lo que se puede realizar utilizando cadenas, podemos plantear diferentes problemas con esto, además de que el uso tan sencillo que fue de utilizar los ciclos de tipo **"while"**, sabiendo, implementándolo de una manera correcta, por la parte cómo saber eliminar algunos de los caracteres para que nuestro programa funcione de manera correcta.