**#Include string.h;**

printf(“%s” cadena[1] );

**FUNCION: STRCPY()**

Copia el contenido de un de un string encima de otro string (lo superpone) , seria el equivalente a escribir :

Texto1 [] = Texto2 [] ;

El contenido de Texto1 queda igual a Texto2 , y Texto2 no se modifica;

**Strcpi( texto , ”Hola”);**

**Indicamos el string que queremos modificar , y su índice( en este caso el 2 ) ;**

**Indicamos que se va a sobreescribir en esa posición , aca SI podemos escribir directamente una cadena de caracteres , pero también podemos poner otro string , o un valor que este en otro array de strings , por ejemplo , strcpi ( texto , saludos[3]);**

**“Texto” pasa a tener el mismo contenido que esta en el array “saludos” , índice 3**

**FUNCION : FGETS()**

Sirve para no excederse en la cantidad de caracteres declarada.

Importante : Los últimos 2 caracteres de un string tomado por fgets , son un carácter especial , y el \0; es decir si ingresamos HOLA , quedaría{ ‘h’ ’o’ ’l’ ’a’ ‘CARÁCTER ESP’ ‘/0 ’}

Por eso en la function ,

Char texto[];

fgets(texto,sizeof(texto)-2,stdin);

sizeof : Valida que se puedan ingresar hasta (tamaño de “texto”) (ahí puede ir un numero) caracteres : restamos los ultimos 2 caracteres del string porque están reservados para un carácter especial del fgets y el \0;

osea , si hiciéramos fgets(texto,25,stdin); podríamos ingresar hasta 23 caracteres.

Stdin :limpia la memoria

**Despues:**

Declaramos una variable “cantidad” que va a guardar la longitud de el string.

**Int cantidad**

cantidad = strlen(nombre); asignamos la longitud de el string a “cantidad “;

PARA QUE? >>>> para cambiar el anteúltimo carácter del string ( que seria el carácter especial del fgets , a un \0 , y asi lo eliminamos):

texto[cantidad-1] = '\0';

**FUNCION : STRLEN()**

Devuelve la cantidad de caracteres que tiene una cadena;solo omitiendo el /0

Sirve para validar .

**FUNCION : STRCMP( , )**

**Int strcmp (texto1,texto2);**

**Strcmp:** devuelve un entero , recibe una cadena y otra cadena , sirve para ordenar alfabéticaente por ejemplo

Return 0 si ambas son iguales ;

Return valor < 0 si la primera es menor a la 2da cadena;

Return valor > 0 si la primera es mayor que la segunda;

Es sensible a mayusc y minúsculas;

FUNCION: **strupr() ; strlwr() ; toUpper () ; toLower()**

Strupr(string) = Pasa todo el string a mayúsculas.

Strlwr(string)= Pasa todo el string a minúsculas.

toUpper() = pasa UN SOLO CARÁCTER a mayúsculas.

toLower() =Pasa UN SOLO CARÁCTER a minúsculas.

Para que sirve , para comparar cadenas sin problemas , para dejar prolijos nombres y apellidos, para al momento de validar ahorrarnos 2 parametros extra (si no ponemos esto , aparte de validar s y n , también deberíamos validar S y N)

**FUNCION strcat( , );**

Toma 2 cadenas de caracteres , y las concatena . Se puede ingresar texto “crudo”;

**Strcat(cadena1 ,cadena2) ; Strcat(cadena1 ,”muy bonito”);**

Sirve para guardar en una sola variable , un string hecho de muchos string , por ejemplo si quiero guardar un apellido conjunto con un nombre.

Para no hacer esto printf(“%s %s” , nombres[2] , apellido [2]);

y si esto printf(“%s” ,nombreyApellido[2])