Nombre de la práctica	CADENAS Y FUNCIONES			No.	12 ~~13~~
Asignatura:	METODOS NUMERICOS	Carrera:	ING. SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	10

I. Competencia(s) específica(s):

II. Lugar de realización de la práctica (laboratorio, taller, aula u otro):

Otro

III. Material empleado:

Dev C++

IV. Desarrollo de la práctica:

Una cadena es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el carácter '\0'.

Ejemplo 1:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char cad[10];
6      int i;
7      printf ("INTRODUCE UNA PALABRA \n"  );
8      gets (&cad);
9      printf ("---%s\n" ,cad);
10
11     system("Pause");
12     return 0;
13 }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Cadena.exe
INTRODUCE UNA PALABRA
Yorely
---Yorely
Presione una tecla para continuar . . .
```

¿Cómo declarar una cadena?

Crearla como arreglo sin tamaño

char cad [] = "Es una cadena";

Ejemplo 2:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main(){
5       char cad[]="Es una cadena" ;
6       printf ("%s\n",cad);
7       system("Pause");
8       return 0;
9   }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Cadena1.exe
Es una cadena
Presione una tecla para continuar . . .
```

Funciones de cadena: strlen

Devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.

strlen(<cadena>)

Ejemplo 3

```
1   #include <stdio.h>
2   #include <stdlib.h>
3
4   int main(){
5       int len;
6       char cad[]="ES UNA CADENA" ;
7       len=strlen(cad);
8       printf ("LA LONGITUD DE:\'%s\'ES:%d\n" ,cad,len);
9       system("Pause");
10      return 0;
11  }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strlen.exe
LA LONGITUD DE:'ES UNA CADENA'ES:13
Presione una tecla para continuar . . .
```

Ejercicio 4:
Escribe un programa que reciba una palabra por teclado.
De acuerdo a la longitud de la palabra (N) que se ingresó por teclado imprime un cuadrado de asteriscos de (N x N).

```c
#include <stdio.h>
#include <stdlib.h>

int main(){
    char cad[]="";
    int len,i,j;
    puts(":3 INGRESE UNA PALABRA:" );
    gets (&cad);
    len =strlen(cad);
    printf ("LA LONGITUD DE LA PALABRA ES %d\n" ,len);
    for( i =0;i <len;i ++){
        for( j =0;j <len;j ++){
            printf ("*");
        }
        puts ("\n");
    }
    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Cuadro.exe
:3 INGRESE UNA PALABRA:
Yorely
LA LONGITUD DE LA PALABRA ES 6
******

******

******

******

******

******

Presione una tecla para continuar . . .
```

Funciones de cadena: strcpy

Copia el contenido de:
<cadena_origen> en <cadena_destino>.

strcpy(<cadena_destino>, <cadena_origen>)

Ejemplo 5:

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    int len;
    char origen[]="Origen";
    char destino[7];
    strcpy(destino,origen);
    printf("destino:%s\n",destino);
    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strcpy.exe
destino:Origen
Presione una tecla para continuar . . .
```

Ejercicio 6:
Escribe un programa que reciba por teclado dos palabras y cada una de ellas las almacene en un arreglo.
Después intercambia sus contenidos. Imprime el antes y el después.

antes	después
Palabra_1 = ' Programacíon'	Palabra_1 = 'Computadora'
Palabra _2= 'Computadora'	Palabra _2= 'Programacíon'

```
#include <stdio.h>
#include <stdlib.h>

int main(){
    char antes[]=" ";
    char despues[]=" ";
    char guardar[]=" ";

    puts("INGRESE UNA PALABRA:\n");
    scanf("%s",&despues);
    puts("INGRESE OTRA PALABRA:\n");
    scanf("%s",&antes);

    strcpy(guardar,despues);
    strcpy(despues,antes);

    printf("\n1 Una palabra:%s\n1 Otra palabra:%s \n",guardar,antes);
    printf("\n2 Una palabra:%s\n2 Otra palabra:%s \n",antes,guardar);

    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Palabra.exe
INGRESE UNA PALABRA:

Yorely
INGRESE OTRA PALABRA:

Anna

1 Una palabra:Yorely
1 Otra palabra:Anna

2 Una palabra:Anna
2 Otra palabra:Yorely
Presione una tecla para continuar . . .
```

Funciones de cadena: strcat

Concatena el contenido de <cadena_origen> al final de <cadena_destino>

strcat(<cadena_destino>, <cadena_origen>)

Ejemplo 7:

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int len;
6      char origen[]="BRISAS";
7      char destino[11]="PARA";
8      strcat(destino,origen);
9      printf("destino:%s\n ",destino);
10     system("Pause");
11     return 0;
12 }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strcat.exe
destino:PARABRISAS
 Presione una tecla para continuar . . .
```

Ejercicio 8:
Escribe un programa que reciba por teclado dos palabras.
Y concatene N veces la segunda palabra a la primer palabra.
Donde N es la longitud de la primera palabra.
Ej.
Palabra1: ’para’
Palabra2: ’brisas‘
Palabra1: parabrisasbrisasbrisasbrisas

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    int len,i;

    char uno[30]=" ";
    char dos[30]=" ";

    puts("INGRESE UNA PALABRA:\n" );
    scanf("%s",&uno);
    puts("INGRESE OTRA PALABRA:\n" );
    scanf("%s",&dos);

    len =strlen(uno);


    printf("\n1 Una palabra:%s\n" ,uno);
    printf("\n2 Otra palabra:%s\n" ,dos);
    printf("\n3 EL TAMAÑO DE LA PALABRA 1 ES:%d \n " ,len);
        for(i=0;i<len;i++){
            strcat (uno,dos);
        }
    printf("%s \n",uno);
    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\OtrasPalabras.exe
INGRESE UNA PALABRA:

para
INGRESE OTRA PALABRA:

brisas

1 Una palabra:para

2 Otra palabra:brisas

3 EL TAMAÑO DE LA PALABRA 1 ES:4
 parabrisasbrisasbrisasbrisas
Presione una tecla para continuar . . .
```

Funciones de cadena: strcmp

Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.
Un número positivo si <cadena1> es menor que <cadena2>
Un número negativo (mayor que cero) si <cadena1> es mayor que <cadena2>.

strcmp(<cadena1>, <cadena2>)

Ejemplo 9:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    int res;

    char str1[]="para" ;
    char str2[]="brisas" ;

    res =strcmp(str1,str2);
    printf("Resultado:%d\n" ,res);

    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strcmp.exe
Resultado:1
Presione una tecla para continuar . . .
```

Ejemplo 10:

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    int res;

    char str1[]="brisas" ;
    char str2[]="para" ;

    res =strcmp(str1,str2);
    printf("Resultado:%d\n" ,res);

    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strcmp.exe
Resultado:-1
Presione una tecla para continuar . . .
```

Ejemplo 11:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #include <string.h>
4   int main(){
5       int res;
6
7       char str1[]="para" ;
8       char str2[]="para" ;
9
10      res =strcmp(str1,str2);
11      printf ("Resultado:%d\n" ,res);
12
13      system("Pause");
14      return 0;
15  }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\Strcmp.exe
Resultado:0
Presione una tecla para continuar . . .
```

Ejercicio 12:
Escribe un programa que reciba por teclado dos palabras y te indique cuál de ellas es mayor y cuál es la menor.
En caso de ser iguales, que imprima la leyenda ’ambas palabras son iguales’.
Compara las dos cadenas y devuelve un 0 si las dos cadenas son iguales.
Un número negativo si <cadena1> es menor que <cadena2>
Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    int res;
    char str1[]=" ";
    char str2[]=" ";
    puts("INGRESE UNA PALABRA:\n");
    scanf("%s",&str1);
    puts("INGRESE OTRA PALABRA:\n");
    scanf("%s",&str2);
    res =strcmp(str1,str2);

    if(res>0||res<0){
        printf("Resultado:%d\n",res);
    }
    else{
    printf("AMBAS PALABRAS SON IGUALES\n");
    printf("Resultado:%d\n",res);
    }
    if(str1>str2){
    printf("la primer palabra es mayor\n",res);
    }
    system("Pause");
    return 0;
}
```

Y:\Documentos\TERCER SEMESTRE\C++\StrcmpEjer.exe

```
INGRESE UNA PALABRA:

perro
INGRESE OTRA PALABRA:

gato
Resultado:1
la primer palabra es mayor
Presione una tecla para continuar . . .
```

Y:\Documentos\TERCER SEMESTRE\C++\StrcmpEjer.exe

```
INGRESE UNA PALABRA:

gato
INGRESE OTRA PALABRA:

gato
AMBAS PALABRAS SON IGUALES
Resultado:0
la primer palabra es mayor
Presione una tecla para continuar . . .
```

Ejercicio 13:
Crea un programa que detecte una palabra palíndroma.
Los palíndromos son palabras que se leen igual de izquierda a derecha que de derecha a izquierda.
Ejemplo: ala, rotor, salas.

```c
#include <stdio.h>
#include <stdlib.h>
int main(){
    int n,x,a=0,p=0,e=0,u;
    char str1[20];
    char str2[20];
    char b="";
    puts("INGRESE UNA PALABRA:\n");
    gets(&str1);
    n=strlen(str1);
    for(u=0;u<=n;u++){
        if(str1[u==b]){
            u++;
        }
        str2[e]=str1[u];
        e++;
    }
    x=0;
    n=strlen(str2);
    for(x=n-1;x>=0;x--){
        if(str2[x]==str2[a]){
            p++;
        }
        a++;
    }
    if(p==n){
        printf("ES UNA PALABRA POLINDROMA\n");
    }else{

        printf("NO ES UNA PALABRA POLINDROMA\n");
    }
    system("Pause");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\PalabraPolindroma.exe
INGRESE UNA PALABRA:

ala
ES UNA PALABRA POLINDROMA
Presione una tecla para continuar . . .
```

```
Y:\Documentos\TERCER SEMESTRE\C++\PalabraPolindroma.exe
INGRESE UNA PALABRA:

gato
NO ES UNA PALABRA POLINDROMA
Presione una tecla para continuar . . .
```

Ejercicio 14:
Crea un programa que cuente cuantas ocurrencias de cada letra contiene una palabra.
Ejemplo:
Palabra
P →1
a →3
l →1
b →1
r →1

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char caracteres[150];
    int palabra[200];
    int i;
    puts("Ingrese una palabra");
    scanf("%s",caracteres);
    for(i=0; i<150; i++){
        if(caracteres[i]<0){
            caracteres[i]=(i*-1);
        }
    }
    for(i=0; i<200; i++){
        palabra[i]=0;
    }
    for(i=0; caracteres[i]!='\0'; i++){
        palabra[caracteres[i]]++;
    }
    for(i=0; i<200; i++){
        if(palabra[i]>0){
            printf("%c = %i\n", i, palabra[i]);
        }
    }
    return 0;
}
```

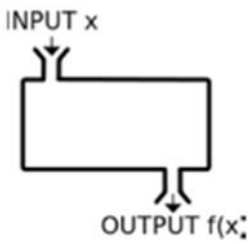
```
Y:\Documentos\TERCER SEMESTRE\C++\prueba.exe
Ingrese una palabra
Palabra
P = 1
a = 3
b = 1
l = 1
r = 1

--------------------------------
Process exited after 6.108 seconds with return value 0
Presione una tecla para continuar . . .
```

FUNCIONES:

Matemáticamente una función es una operación que toma uno o más valores llamados argumentos y produce un valor llamado resultado.

Una función es un bloque de código reconocido por un identificador que realiza un trabajo específico.
Su propósito es dividir los programas en módulos manejables separados (divide y vencerás).



Ventajas:

1. Facilita el diseño descendente.
2. Los procedimientos dentro de ellas se pueden ejecutar varias veces.
3. Facilita la división de tareas.
4. Se pueden probar individualmente
5. Con funciones apropiadamente diseñadas, es posible ignorar como se realiza una tarea, sabiendo qué es lo que hacen.

Modo de uso

1. Funciones diseñadas para realizar operaciones a partir de sus argumentos y devolver un valor basado en sus cálculos.
2. Funciones que no reciben argumentos, realizan un proceso y devuelven un valor.
3. Funciones que no tienen argumentos ni valor de retorno explícito, operan sobre el entorno de variables globales o atributos del sistema operativo.

```
int suma(int a, int b){
    int c;
    c=a+b;
     return c;
}
```

```
flooat promedio(int a, int b, int c){
        int suma=a+b+c;
        float prom=sum/3;
        return prom;
}
```

Declaración de funciones:

El formato para la declaración de funciones es:

```
tipo nombre_funcion(lista de parámetros){
        cuerpo de la función
}
```

tipo: especifica el tipo de valor que devuelve la función. Si no se especifica tipo, el compilador asume que es entero (int).

lista de parámetros : es la lista de nombres de variables separados por comas con sus tipos asociados que reciben los valores de los argumentos actuales de la llamada a la función.

```
tipo nombre_funcion(lista de parámetros){
      cuerpo de la función
}
```

° Forza la salida inmediata de la función en que se encuentra.
° Una función puede retornar valor sólo cuando el tipo de retorno no es void.
° Devuelve un valor a la función que realizó la llamada.
return (expresión);

° Tradicionalmente en C se declaran como prototipos al inicio del programa.
° Después se declara la función main, y después se hace la declaración formal de las funciones.
° También pueden declararse las funciones al inicio del programa y después declarar la función main sin declarar prototipo.

int suma(int a, int b){
int c;
return int
c=a+b;
}
	flooat promedio(int a, int b, int c){
	int suma=a+b+c;
	float prom=sum/3;
	
	}

Llamadas a funciones:

Para llamar a una función se especifica su nombre y la lista de argumentos sin poner el tipo de dato.

nombre_funsion(var1,var2,…..varN)

En una llamada habrá un argumento real por cada argumento formal, respetando el orden de declaración.
Argumento formal: Los que aparecen en la definición de la función.
Argumento real: Los que se pasan en la llamada a la función.

Paso de parámetros por valor

° Se hace una copia del valor del argumento en el parámetro formal.
° La función opera internamente con estos últimos.
° Los parámetros formales se crean al entrar a la función y se destruyen al salir de ella, cualquier cambio realizado por la función en los parámetros formales no tienen ningún efecto sobre los argumentos.

Ejemplo 15:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   float  promedio (float  a,float  b);
4   int  main(){
5         float  a=5,b=10,prom;
6         prom=promedio (a,b);
7         printf ("El promedio es:%2.1f\n"
8         system("Pause");
9         return  0;
10  }
11  float  promedio  (float  a,float  b){
12        float  prom;
13        a=a+3;
14        b=b+3;
15        prom=(a+b)/2;
16
17        return  prom;
18  }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcion.exe
El promedio es:10.5
Presione una tecla para continuar . . .
```

Variables locales y globales

° Variables Locales:

Se declaran dentro de la función y sólo están disponibles durante su ejecución.

Se crean cuando se entra en ejecución una función y se destruyen cuando se termina.

° Variables globales:

Se declaran fuera de las funciones. Pueden ser utilizadas por todas las funciones.

Existen durante toda la vida del programa.

Ejercicio 16:

° Escribir una función que se llame maximo que reciba dos número por parámetros y que regrese el mayor de ellos.

```c
#include <stdio.h>
#include <stdlib.h>
int maximo(int num1,int num2);
int main(){
    int a=100;
    int b=200;
    int resultado;
    resultado=maximo(a,b);
    printf("EL NUMERO MAYOR ES %d\n",resultado);
    return 0;
}
int maximo (int num1,int num2){
    int resultado;
    if(num1>num2){
        resultado=num1;
    }else{
        resultado=num2;
    }
    return resultado;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcion1.exe
EL NUMERO MAYOR ES 200

--------------------------------
Process exited after 0.02038 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 17:

Escribir una función que reciba caracteres del teclado hasta recibir un espacio o un salto de línea (enter) y a continuación mostrar todos los caracteres en orden inverso.

Ejemplo:

Entrada: Hola

Salida: aloH

```c
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
int main(){
    char a[200];
    int len,i=0,j;
    puts("INGRESE UNA PALABRA:\n");
    scanf("%s",a);
    printf("LA PALABRA INTRODUCIDA ES: %s\n",a);
    len=strlen(a);
    while(a[i]!='\0'){
        i++;
    }
    printf("LA PALABRA A LA REVES ES: ");
    for(j=i-1;j>=0;j--){
        printf("%c",a[j]);
    }
    puts("");
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcion2.exe
INGRESE UNA PALABRA:

Hola
LA PALABRA INTRODUCIDA ES: Hola
LA PALABRA A LA REVES ES:  aloH

--------------------------------
Process exited after 4.021 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio 18:
Escribir una función que tome como parámetros las longitudes de los tres lados de un triángulo (a, b, c) y devuelva el área del triángulo.

$$A = \sqrt{p(p-a)(p-b)(p-c)}\ D \qquad p = \frac{a+b+c}{2}$$

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
float calcular(float a,float b,float c){
    if(a!=0&&b!=0&&c!=0){
        float s=(float)(a+b+c)/2;
        return s;
    }
}
float area(float a,float b, float c){
    if(a!=0&&b!=0&&c!=0){
        float s=calcular(a,b,c);
        float area=0;
        if(s>a&&s>b&&s>c){
            area=sqrt(s*(s-a)*(s-b)*(s-c));
        }
        return area;
    }
}
int main(){
    printf("AREA DEL TRIANGULO\n");
    int i=0;
    float lados[3];
    for(i=0;i<3;i++){
        printf("INGRESA LA LONGIUD DEL LADO %i:\n",i+1);
        scanf("%f",&lados[i]);
    }
    float area=calcular(lados[0],lados[1],lados[2]);
    if(area==0){
        printf("LOS DATOS INGRESADOS NO CORRESPONDEN");
    }else{
        printf("EL AREA DEL TRIANGULO ES:%.3f",area);
    }
    getch();
    return 0;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcion3.exe
AREA DEL TRIANGULO
INGRESA LA LONGIUD DEL LADO 1:
5
INGRESA LA LONGIUD DEL LADO 2:
4
INGRESA LA LONGIUD DEL LADO 3:
3
EL AREA DEL TRIANGULO ES:6.000
```

Funciones recursivas

° Se llaman funciones recursivas a aquellas que se llaman a sus mismas de forma repetida hasta que se cumpla alguna condición.

° Cada llamada implica el almacenamiento de variables de estado y otros parámetros.

Ejemplo 19:

```
1   #include <stdio.h>
2   #include <stdlib.h>
3   #include <string.h>
4   int main(){
5       int x,y,max;
6       x=2;
7       y=3;
8       max=potencia(x,y);
9       printf("LA POTENCIA ES: %d\n",max);
10      system("Pause");
11      return 0;
12  }
13  int potencia (a,b){
14      if(b<1)
15      return 1;
16      return a*potencia(a,b-1);
17  }
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcionRec.exe
LA POTENCIA ES: 8
Presione una tecla para continuar . . .
```

Ejercicio 20:

Haz un programa con funciones recursivas que calcule el factorial de un número n ingresado desde teclado.

Ej. N = 5

5!= 4! * 5
4! = 3! * 4
3! = 2! * 3
2!= 1! * 2
1!= 0! * 1
0!= 1

```c
#include <stdio.h>
#include <stdlib.h>
#include<string.h>
int factorial(int numero);
int main(){
    int numero,resul;
    puts("INTRODUCE UN NUMERO");
    scanf("%d",&numero);
    resul=factorial(numero);
    printf("EL FATORIAL DE %d ES: %d\n",numero,resul);
    return 0;
}
int factorial(int numero){
    int resul;
    resul=1;
    while(numero>1){
        resul*=numero;
        numero--;
        }
    return resul;
}
```

```
Y:\Documentos\TERCER SEMESTRE\C++\funcionRec1.exe
INTRODUCE UN NUMERO
8
EL FATORIAL DE 8 ES: 40320


--------------------------------
Process exited after 3.126 seconds with return value 0
Presione una tecla para continuar . . .
```