

Nombre de la práctica	Ejercicios en Lenguaje C			No.	4
Asignatura:	Métodos Numéricos	Carrera:	ISIC	Duración de la práctica (Hrs)	

Nombre: Rodrigo Javier Martínez

Grupo: 3041

I. Competencia(s) específica(s):

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

- Aula y casa

III. Material empleado:

- Dev-C++
- Word

IV. Desarrollo de la práctica:

Esta es la continuación del curso de Lenguaje C que llevamos a cabo en Métodos Numéricos, en donde estamos realizando ejercicios en este lenguaje.

• Cadenas

Para comenzar vamos a ver que es una cadena, es un arreglo de caracteres. En donde, por lo general el último elemento deberá ser el carácter '\0'.

¿Cómo declarar una cadena?

```

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

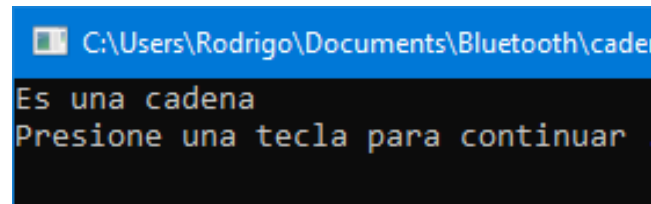
```

C:\Users\Rodrigo\Documents\Ejercicios C\cadena.exe
Introduce una palabra
Hola Mundo
Hola MundoPresione una tecla para continuar
```

Crear una cadena es como crear un arreglo sin tamaño:

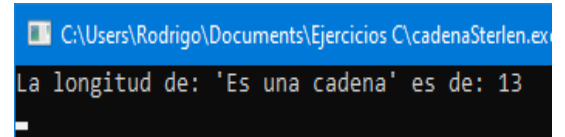
```
char cad [ ] = "Es una cadena";
```

```
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  }
```



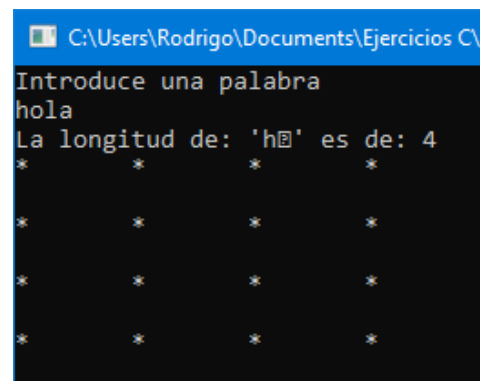
Funciones de cadena: strlen devuelve la longitud de la cadena sin tomar en cuenta el carácter de final de cadena.

```
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
9      printf("La longitud de: \'%s\' es de: %d\n", cad, len);
10     gets(&cad);
11     printf("%s", cad);
12     system("pause");
13     return 0;
14 }
```



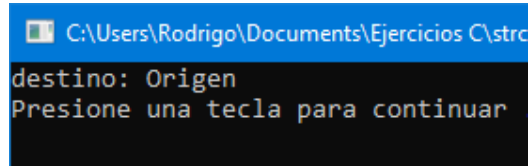
Escribe un programa que reciba una palabra por teclado. De acuerdo a la longitud de la palabra (N) que se ingresó por teclado imprime un cuadrado de asteriscos de (N x N).

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main () {
5      int i;
6      int j;
7      int len;
8      char cad[]="";
9      int oper;
10     printf("Introduce una palabra \n");
11     gets(&cad);
12     len = strlen(cad);
13     printf("La longitud de: \'%s\' es de: %d\n", cad, len);
14     for(i=1;i<=len;i++){
15         for(j=1;j<=len;j++){
16             printf("*\t");
17         }
18         printf("\n\n");
19     }
20
21     system("pause");
22     return 0;
23 }
24 }
```



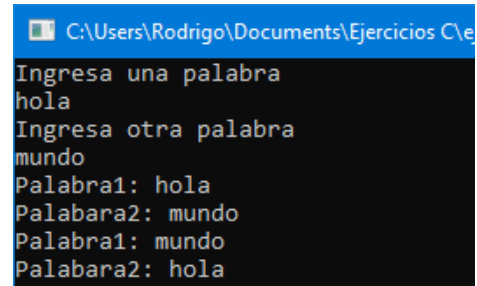
Funciones de cadena: strcpy copia el contenido de: <cadena_origen> en <cadena_destino>.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (){
5     int len;
6     char origen[]="Origen";
7     char destino[7];
8     strcpy(destino, origen);
9     printf("destino: %s ", destino);
10
11     system("pause");
12     return 0;
13 }
```



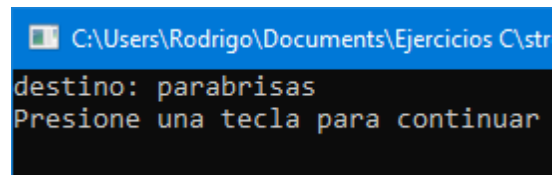
Escribe un programa que reciba por teclado dos palabras y cada una de ellas las almacene en un arreglo. Después intercambia sus contenidos. Imprime el antes y el después

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main (){
5     char antes[]=" ";
6     char despues[]=" ";
7     char nuevo[]=" ";
8
9     puts("Ingresa una palabra");
10    scanf("%s",&despues);
11    puts("Ingresa otra palabra");
12    scanf("%s",&antes);
13
14    strcpy(nuevo,despues);
15    strcpy(despues,antes);
16
17    printf("Palabra1: %s\nPalabara2: %s\n", nuevo, antes);
18    printf("Palabra1: %s\nPalabara2: %s\n", antes, nuevo);
19
20    system("pause");
21    return 0;
22 }
```



Funciones de cadena: strcat concatena el contenido de <cadena_origen> al final de <cadena_destino>

```
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 ", destino);
10    system("pause");
11    return 0;
12
13 }
```





Escribe un programa que reciba por teclado dos palabras. Y concatene N veces la segunda palabra a la primera palabra, donde N es la longitud de la primera palabra.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int len;
6     puts("Ingresa la primera palabra");
7     char palabra1[30]="";
8     scanf("%s", &palabra1);
9
10    puts("Ingresa la primera palabra");
11    char palabra2[30]="";
12
13    scanf("%s", &palabra2);
14    len=strlen(palabra1);
15    int i;
16    for(i=0;i<len;i++){
17        strcat(palabra1, palabra2);
18    }
19    printf("destino: %s \n", palabra1);
20
21    system("pause");
22    return 0;
23
24 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio1
Ingresa la primera palabra
hola
Ingresa la primera palabra
mundo
destino: holamundomundomundomundo
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 negativo si <cadena1> es menor que <cadena2>

Un número positivo (mayor que cero) si <cadena1> es mayor que <cadena2>.

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

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio2
Resultado: 1
Presione una tecla para continuar
```

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

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio3
Resultado: -1
Presione una tecla para continuar
```



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

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio1.exe
Resultado: 0
Presione una tecla para continuar
```

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>.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main( ){
5      int res;
6      puts("Ingresa la primera palabra");
7      char str1[]="";
8      scanf("%s",str1);
9      puts("Ingresa la segunda palabra");
10     char str2[]="";
11     scanf("%s",str2);
12     res = strcmp (str1, str2);
13     if(res == 1){
14         printf("La primera palabra es mayor\n");
15     }else if(res == -1){
16         printf("La segunda palabra es menor\n");
17     }else if(res == 0){
18         printf("Ambas palabras son iguales");
19     }
20     system("pause");
21     return 0;
22 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio2.exe
Ingresa la primera palabra
sistemas
Ingresa la segunda palabra
meca
La segunda palabra es menor
Presione una tecla para continuar
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio3.exe
Ingresa la primera palabra
hola
Ingresa la segunda palabra
hola
Ambas palabras son iguales
Presione una tecla para continuar .
```



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.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main( ){
5      int res;
6      int i,j;
7
8      puts("Ingresa la palabra");
9      char palabra[20];
10     fflush(stdout);
11     gets(palabra);
12     j=strlen(palabra)-1;
13     for(i=0;i<strlen(palabra)/2;i++,j--){
14         printf("%c==%c \n", *(palabra+i), *(palabra+j));
15         if(*(palabra+i)== *(palabra+j)){
16             printf("Es una palabara palindroma\n");
17         }else{
18             printf("No es una palabara palindroma\n");
19         }
20     }
21 }
22
23 system("pause");
24 return 0;
25 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\
Ingresa la palabra
rotor
r==r
Es una palabara palindroma
o==o
Es una palabara palindroma
```

Crea un programa que cuente cuantas ocurrencias de cada letra contiene una palabra.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  void main(){
4      int caracter[200];
5      char cadena[200];
6      int i;
7      printf ("Ingresa una palabra: ");
8      fflush(stdout);
9      gets(cadena);
10
11     for (i=0;i<200;i++){
12         if (cadena[i]<0)
13             cadena[i]=(i*-1);
14     }
15     printf("\n");
16     for (i=0;i<200;i++)
17         caracter[i]=0;
18     for (i=0;cadena[i] != '\0' ;i++)
19         caracter[cadena[i]]++;
20     for (i=0;i<200;i++)
21         if (caracter[i] > 0)
22             printf ("%c = %i\n", i, caracter[i]);
23     printf ("\n");
24     system ("pause");
25     return 0;
26 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ocurrenciasExamen.exe
Ingresa una palabra: palabra
a = 3
b = 1
l = 1
p = 1
r = 1
Presione una tecla para continuar . . .
```

• Funciones

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).

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.

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.
- 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.
- Entre llaves se encuentra el cuerpo de la función.
- Forzar 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 (expresion);
- Se declara la función main, y después se hace la declaración formal de las funciones.
- Pueden declararse las funciones al inicio del programa y después declarar la función main sin declarar prototipo.

Llamadas a funciones

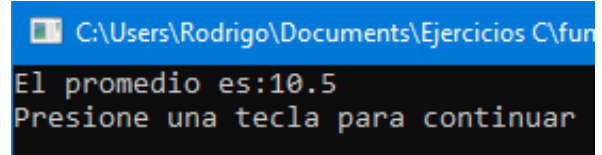
- Para llamar a una función se especifica su nombre y la lista de argumentos sin poner el tipo de dato.
- En una llamada habrá un argumento real por cada argumento formal, respetando el orden de declaración.
 - Argumento formal: Los que aparecen en la definición de la función.
 - Argumento real: Los 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 de un programa con funciones:

```
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",prom);
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 }
```



```
C:\Users\Rodrigo\Documents\Ejercicios C\fun
El promedio es:10.5
Presione una tecla para continuar
```

Variables locales y globales

Variables Locales:

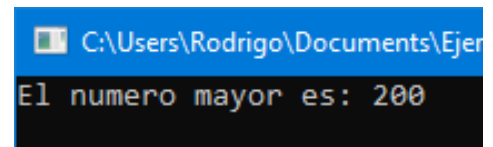
- Se declaran dentro de la función y sólo están disponibles durante su ejecución.
- Se crean cuando se entra en ejecución una función y se destruyen cuando se termina.

Variables globales:

- Se declaran fuera de las funciones. Pueden ser utilizadas por todas las funciones.
- Existen durante toda la vida del programa.

Escribir una función que se llame máximo que reciba dos números por parámetros y que regrese el mayor de ellos.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 int maximo(int num1,int num2);
4 int main(){
5     int a=100;
6     int b=200;
7     int resultado;
8     resultado=maximo(a,b);
9     printf("El numero mayor es: %d\n",resultado);
10    return 0;
11 }
12 int maximo (int num1,int num2){
13     int resultado;
14     if(num1>num2){
15         resultado=num1;
16     }else{
17         resultado=num2;
18     }
19     return resultado;
20 }
```



```
C:\Users\Rodrigo\Documents\Ejer
El numero mayor es: 200
```




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.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int main(){
5      char a[200];
6      int len,i=0,j;
7      puts("Ingresa una palabra:\n");
8      scanf("%s",a);
9      printf("La palabra es: %s\n",a);
10     len=strlen(a);
11     while(a[i]!='\0'){
12         i++;
13     }
14     printf("La palabra alreves es: ");
15     for(j=i-1;j>=0;j--){
16         printf("%c",a[j]);
17     }
18     puts("");
19     return 0;
20 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\fu
Ingresa una palabra:
hola
La palabra es: hola
La palabra alreves es: aloh
```

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.

$$A = \sqrt{p(p-a)(p-b)(p-c)} \quad \text{Donde } p = \frac{a+b+c}{2}$$

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <math.h>
4  float calcular(float a,float b,float c){
5      if(a!=0&b!=0&c!=0){
6          float s=(float)(a+b+c)/2;
7          return s;
8      }
9  }
10 float area(float a,float b, float c){
11     if(a!=0&b!=0&c!=0){
12         float s=calcular(a,b,c);
13         float area=0;
14         if(s>a&s>b&s>c){
15             area=sqrt(s*(s-a)*(s-b)*(s-c));
16         }
17         return area;
18     }
19 }
20 int main(){
21     printf("***Area de un triangulo***\n");
22     int i=0;
23     float lados[3];
24     for(i=0;i<3;i++){
25         printf("Ingresa la medida de un lado %i:\n",i+1);
26         scanf("%f",&lados[i]);
27     }
28     float area=calcular(lados[0],lados[1],lados[2]);
29     if(area==0){
30         printf("Los datos ingresados son");
31     }else{
32         printf("El area del triangulo es:%f",area);
33     }
34     getch();
35     return 0;
36 }
37 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\funci
***Area de un triangulo***
Ingresa la medida del lado 1:
3
Ingresa la medida del lado 2:
6
Ingresa la medida del lado 3:
4
El area del triangulo es:6.500000
```

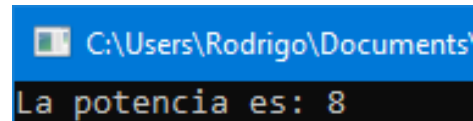
Funciones recursivas

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

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

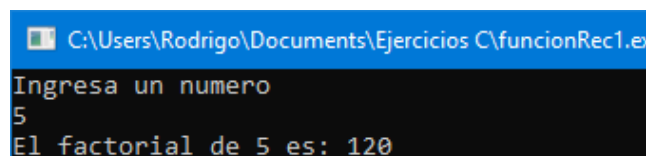
Ejemplo de un programa con Funciones recursivas:

```
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 }
```



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

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  int factorial(int numero);
5  int main(){
6      int numero,resul;
7      puts("Ingresa un numero");
8      scanf("%d",&numero);
9      resul=factorial(numero);
10     printf("El factorial de %d es: %d\n",numero,resul);
11     return 0;
12 }
13 int factorial(int numero){
14     int resul;
15     resul=1;
16     while(numero>1){
17         resul*=numero;
18         numero--;
19     }
20     return resul;
21 }
```



- **Estructuras**

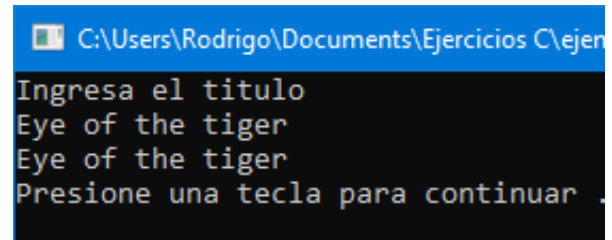
Vamos a ver que es una estructura de datos, está compuesta de elementos individuales que pueden ser de distinto tipo.

Cada uno de los elementos de una estructura se denomina miembro.

Los miembros pueden ser cualquier tipo excepto void.

Ejemplo de cómo realizar una estructura mediante el llenado de datos por teclado.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct CD{
4      char titulo[100];
5      char artista [50];
6
7  };
8  int main(void){
9      struct CD cd1;
10     printf("Ingresa el titulo\n");
11     gets(cd1.titulo);
12
13     printf("%s\n", cd1.titulo);
14
15     system("pause");
16     return 0;
17 }
```



Modifica la estructura para que se pueda almacenar también. Numero de canciones, Año y Precio

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  struct CD{
4      char titulo[100];
5      char artista [50];
6      char numCancion[100];
7      char anio[100];
8      char precio[100];
9  };
10 int main(void){
11     struct CD cd1;
12     printf("Ingresa el titulo\n");
13     gets(cd1.titulo);
14
15     printf("Ingresa nombre del artista\n");
16     gets(cd1.artista);
17
18     printf("Ingresa numero de la cancion\n");
19     gets(cd1.numCancion);
20
21     printf("Ingresa anio\n");
22     gets(cd1.anio);
23
24     printf("Ingresa precio\n");
25     gets(cd1.precio);
26     printf("%s %s %s %s %s\n", cd1.titulo, cd1.artista, cd1.numCancion, cd1.anio, cd1.precio);
27
28     system("pause");
29     return 0;
30 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio1Estructuras.exe
Ingresa el titulo
We are the champions
Ingresa nombre del artista
Queen
Ingresa numero de la cancion
33
Ingresa anio
1998
Ingresa precio
250
We are the champions Queen 33 1998 250
Presione una tecla para continuar . . .
```

Realiza un programa en C que almacene y muestre la información de un empleado de la empresa DATACIC en una estructura llamada “empleado”. La información con que se cuenta del empleado es: nombre, sexo y sueldo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Empleado{
5      char nombre[100];
6      char sexo[50];
7      char sueldo[100];
8  };
9  int main(void){
10     struct Empleado cd1;
11     printf("Ingresa el nombre\n");
12     gets(cd1.nombre);
13
14     printf("Ingresa el sexo\n");
15     gets(cd1.sexo);
16
17     printf("Ingresa el sueldo\n");
18     gets(cd1.sueldo);
19
20     printf("%s %s %s\n", cd1.nombre, cd1.sexo, cd1.sueldo);
21
22     system("pause");
23     return 0;
24
25 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejercicio1Estructuras.exe
Ingresa el nombre
Juan
Ingresa el sexo
Masculino
Ingresa el sueldo
30000
Juan Masculino 30000
Presione una tecla para continuar
```

Realiza un programa en C que almacene y muestre la información de un atleta (deporte, nombre, país, número de medallas).



```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct atomo{
5      int hidrogeno;
6      int oxigeno;
7      char nombre[15]
8  };
9
10 int main(){
11     struct atomo agua[4];
12     agua[0].hidrogeno=2;
13     agua[0].oxigeno=1;
14     gets(agua[0].nombre);
15
16     printf("%s\n", agua[0].nombre);
17
18     system("pause");
19     return 0;
20 }
```

```
C:\Users\Rodrigo\Documents\Bluetooth\arreg
Hola
Hola
Presione una tecla para continuar
```

Arreglos de estructuras, es muy similar a las estructuras simples solo que en este caso se declaran como un arreglo con un tamaño definido.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct Atleta{
5      char deporte[100];
6      char nombre[100];
7      char pais[100];
8      char numMedallas[100];
9  };
10
11 int main(void){
12
13     struct Atleta cd1;
14     printf("Ingresa el deporte\n");
15     gets(cd1.deporte);
16
17     printf("Ingresa el nombre\n");
18     gets(cd1.nombre);
19
20     printf("Ingresa el pais\n");
21     gets(cd1.pais);
22
23     printf("Ingresa el numero de medallas\n");
24     gets(cd1.numMedallas);
25
26     printf("%s %s %s %s\n", cd1.deporte, cd1.nombre,
27         cd1.pais, cd1.numMedallas);
28
29     system("pause");
30     return 0;
31 }
```

```
C:\Users\Rodrigo\Documents\Ejercicios C\ejer
Ingresa el deporte
Futbol
Ingresa el nombre
Juan Perez
Ingresa el pais
Mexico
Ingresa el numero de medallas
3
Futbol Juan Perez Mexico 3
Presione una tecla para continuar
```



Realiza un programa en C que almacene y muestre la información de 5 empleado de la empresa DATACIC en una estructura llamada “empleado”. La información con que se cuenta del empleado es: nombre, sexo y sueldo. Muestra el empleado de menos sueldo.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 struct empleado {
5     char nombre[100];
6     char sexo[100];
7     int salario;
8 };
9 void main(){
10     struct empleado emp[100];
11     int menor_empleado;
12     int menor=500;
13     for(int i=0; i<5; i++){
14         printf("\n");
15         printf("Nombre[%d]: ",i+1);
16         scanf("%s",emp[i].nombre);
17         printf("Sexo[%d]: ",i+1);
18         scanf("%s",emp[i].sexo);
19         printf("Salario[%d]: ",i+1);
20         scanf("%d",&emp[i].salario);
21         if(emp[i].salario<menor){
22             menor=emp[i].salario;
23             menor_empleado=i;
24         }
25     }
26     printf("%s es el trabajador con MENOR salario\n",
27     emp[menor_empleado].nombre);
28 }
```

```
C:\Users\Rodrigo\Documents\Bluetooth\estruc_5emple

Nombre[1]: Jose
Sexo[1]: Masculino
Salario[1]: 3400

Nombre[2]: Maria
Sexo[2]: Femenino
Salario[2]: 4000

Nombre[3]: Ignacio
Sexo[3]: Masculino
Salario[3]: 2500

Nombre[4]: Teresa
Sexo[4]: Femenino
Salario[4]: 3200

Nombre[5]: Juan
Sexo[5]: Masculino
Salario[5]: 2700
Jose es el trabajador con MENOR salario
```

Realiza un programa en C que almacene y muestre la información de 10 atleta (deporte, nombre, país, numero de medallas) y te diga que atleta tiene mayor número de medallas.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4 struct datos{
5     char nombre[40];
6     char pais[25];
7 };
8 struct atleta{
9     char deporte[30];
10    int nmedallas;
11    struct datos pers;
12 };
13 struct atleta ats[10];
14 void main(){
15     int a,posmay,maynmed=0;
16     for(a=0; a<10; a++){
17         printf("\n");
18         printf("Nombre[%d]: ",a+1);
19         scanf("%s",ats[a].pers.nombre);
20         printf("Pais[%d]: ",a+1);
21         scanf("%s",ats[a].pers.pais);
22         printf("Deporte[%d]: ",a+1);
23         scanf("%s",ats[a].deporte);
24         printf("Numero de Medallas[%d]: ",a+1);
25         scanf("%d",&ats[a].nmedallas);
26     }
```

```
27     for(a=1;a<10;a++){
28         if(ats[a].nmedallas>maynmed){
29             posmay=a;
30             maynmed=ats[a].nmedallas;
31         }
32     }
33     printf("\n\t****DATOS DEL ATLETA CON MAYOR NUMERO DE MEDALLAS****\n");
34     printf("NOMBRE: %s\n",ats[posmay].pers.nombre);
35     printf("PAIS: %s\n",ats[posmay].pers.pais);
36     printf("DEPORTE: %s\n",ats[posmay].deporte);
37     printf("NUMERO DE MEDALLAS: %d\n",ats[posmay].nmedallas);
38 }
```



```
C:\Users\Rodrigo\Documents\Bluetooth\estr

Nombre[1]: Juan
Pais[1]: Mexico
Deporte[1]: Atletismo
Numero de Medallas[1]: 3

Nombre[2]: Olga
Pais[2]: Colombia
Deporte[2]: Tiro
Numero de Medallas[2]: 4

Nombre[3]: Jhon
Pais[3]: EU
Deporte[3]: Pesas
Numero de Medallas[3]: 2

Nombre[4]: Luis
Pais[4]: Mexico
Deporte[4]: Clavados
Numero de Medallas[4]: 7

Nombre[5]: Helen
Pais[5]: España
Deporte[5]: Equitacion
Numero de Medallas[5]: 3
```

```
Nombre[6]: Isabel
Pais[6]: Portugal
Deporte[6]: Boxeo
Numero de Medallas[6]: 2

Nombre[7]: Noe
Pais[7]: Africa
Deporte[7]: Atletismo
Numero de Medallas[7]: 7

Nombre[8]: Maria
Pais[8]: Mexico
Deporte[8]: Nado
Numero de Medallas[8]: 9

Nombre[9]: Jorge
Pais[9]: Peru
Deporte[9]: Lanzamiento
Numero de Medallas[9]: 4

Nombre[10]: Merry
Pais[10]: Inglaterra
Deporte[10]: Karate
Numero de Medallas[10]: 1

****DATOS DEL ATLETA CON MAYOR NUMERO DE MEDALLAS****
NOMBRE: Maria
PAIS: Mexico
DEPORTE: Nado
NUMERO DE MEDALLAS: 9
```

Envío de estructuras a funciones:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct fraccion{
4     int den;
5     int num;
6
7 };
8 int main(void){
9     struct fraccion frac1;
10    struct fraccion frac2;
11    frac1.den = 3;
12    frac1.num = 4;
13    frac2.den = 6;
14    frac2.num = 2;
15    multipli(frac1, frac2);
16    system("pause");
17    return 0;
18 }
19 int multipli(struct fraccion mul1, struct fraccion mul2){
20     int den = mul1.den*mul2.den;
21     int nom = mul1.num*mul2.num;
22
23     printf("%d/%d\n", den, nom);
24     return 0;
25 }
```

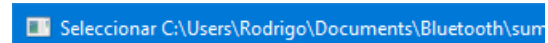
```
C:\Users\Rodrigo\Documents\Bluetooth\mul
18/8
Presione una tecla para continuar
```

Implementar funciones para la suma, la resta y la división de fracciones usando la estructura fracción.

$$\text{Suma: } \frac{a}{b} + \frac{c}{d} = \frac{(d * a) + (b * c)}{b * d}$$

$$\text{División } \frac{a}{b} \div \frac{c}{d} = \frac{a * d}{b * c}$$

```
1 #include <stdio.h>
2 #include <stdlib.h>
3 struct fraccion{
4     int num1;
5     int num2;
6     int num3;
7     int num4;
8 };
9 int main(){
10     struct fraccion in;
11     double suma;
12     double division;
13     puts("Ingresa un numero\n");
14     scanf("%d", &in.num1);
15     puts("Ingresa otro numero\n");
16     scanf("%d", &in.num2);
17     puts("Ingresa otro numero\n");
18     scanf("%d", &in.num3);
19     puts("Ingresa otro numero\n");
20     scanf("%d", &in.num4);
21
22     suma = (in.num4*in.num1)+(in.num2*in.num3)/(in.num2*in.num4);
23     division = (in.num1*in.num4)/(in.num2*in.num3);
24     printf("El resultado la suma es %f\n", suma);
25     printf("El resultado de la division es %f\n", division);
26
27     system("pause");
28     return 0;
29 }
```



```
Ingresa un numero
5
Ingresa otro numero
1
Ingresa otro numero
7
Ingresa otro numero
9
El resultado la suma es 45.000000
El resultado de la division es 6.000000
Presione una tecla para continuar . . .
```

V. Conclusiones:

Para concluir podemos decir que el lenguaje de programación C es uno de los más completos y complejos de utilizar ya que cuenta con muchas funciones las cuales nos pueden ayudar a resolver distintos problemas ya que los puedes implementar en la programación y sacarle provecho a esta. También agregar que el lenguaje C es la base casi toda la programación que su estructura es muy similar a las de los demás lenguajes.