

Nombre de la práctica	EJERCICIOS EN CLASE			No.	5
Asignatura:	METODOS NUMERICOS	Carrera:	ING. EN SISTEMAS COMPUTACIONALES	Duración de la práctica (Hrs)	

NOMBRE DEL ALUMNO: ALEXIS LUZ PEDRAZA

GRUPO: 341

## I. MATERIAL EMPLEADO:

- Dev C++
- Word
- Libreta

## II. DESARROLLO DE LA PRACTICA:

### CADENAS

En este ejemplo podemos ver como ingresar una cadena de texto en un arreglo desde teclado, en este caso se guarda en el arreglo de tipo char "cad" con un tamaño de 10, se captura la cadena con un "scanf" y se imprime en pantalla.

```
int main(){
    char cad[10];
    int i;
    printf("Introduce una palabra \n");
    scanf("%s", cad);
    printf("%s", cad);
    system("Pause");
    return 0;
}
```

Este ejemplo es el mismo que el anterior, solo que este tiene una mejora, se cambia el “scanf” por el “gets”, el “gets” lo que hace es capturar una cadena más larga, con todo y espacios, por ejemplo, “Es una cadena”, al momento de imprimirse, lo va a imprimir todo, no solo la primera palabra.

```
int main(){  
    → char cad[10];  
    printf("Introduce una palabra \n");  
    gets (cad);  
    printf("\n %s ", cad);  
    system("Pause");  
    return 0;  
}
```

Este ejemplo muestra cómo se puede ingresar directamente la cadena al arreglo sin necesidad de pedirlo desde consola.

```
int main(){  
    → char cad[]="Es una cadena";  
    printf("%s \n", cad);  
    system("Pause");  
    return 0;  
}
```

## STRLEN

Este ejemplo muestra cómo sacar la longitud de una cadena, en este caso, el arreglo “cad” tiene la cadena “Es una cadena”, se saca la longitud y se guarda en la variable de tipo entero “len”, después, se imprime en pantalla su longitud, que es de 13.

```
int main(){  
→ int len;  
  char cad[]="Es una cadena";  
  len = strlen(cad);  
  printf("La longitud de:'%s\' es:  
%d\n", cad, len);  
  system("Pause");  
  return 0;  
}
```

## STRCPY

En este ejemplo, se hace uso del “strcpy”, se crean 2 variables de tipo char, “origen” con la cadena “Origen” y “destino” con el tamaño de 7, lo que hace el “strcpy”, es que intercambia de lugar las cadenas, pasa de un arreglo a otro, en este caso, “Origen” está en el arreglo “origen” y usando el “strcpy”, se pasara al arreglo “destino” y se imprimirá en pantalla.

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

Este es un ejercicio del “strcpy”, pedimos 2 palabras desde consola para intercambiarlas después con el “strcpy”.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      char origen[10];
6      puts("INGRESA LA PRIMERA PALABRA");
7      scanf("%s", &origen);
8      char destino[20];
9      puts("INGRESA LA SEGUNDA PALABRA");
10     scanf("%s", &destino);
11     puts("\n    ANTES");
12     printf("Palabra 1: '%s' \n", origen);
13     printf("Palabra 2: '%s' \n", destino);
14     puts("\n    DESPUES");
15     printf("Palabra 1: '%s' \n", destino);
16     printf("Palabra 2: '%s' \n", origen);
17
18     system("Pause");
19     return 0;
20 }
```

En este caso, la primera palabra es “HOLA” y la segunda es “MUNDO”, imprimimos el antes y el después para verificar que si se había hecho el cambio de cadenas.

```
INGRESA LA PRIMERA PALABRA
HOLA
INGRESA LA SEGUNDA PALABRA
MUNDO

    ANTES
Palabra 1: 'HOLA'
Palabra 2: 'MUNDO'

    DESPUES
Palabra 1: 'MUNDO'
Palabra 2: 'HOLA'
Presione una tecla para continuar . . .
```

## **FUNCIONES**

Este ejemplo muestra cómo funcionan las funciones, se crea aparte del método main, se crea una función llamada "promedio", con 2 argumentos de tipo "float", "a" y "b", estos valores ya se les colocho un valor en el método main, "a" con valor de 5 y "b" con valor de 10, en la función se va a hacer la suma de 3 en cada variable, después, esas 2 variables se van a sumar y se dividirá entre 2, el resultado se guardara en la variable "prom" y se ira al método main para que haga la impresión de pantalla.

```
float promedio (float a, float b);  
int main(){  
    float a = 5, b=10 , prom;  
    prom = promedio (a, b);  
    printf ("El promedio es: %2.1f\n", prom);  
    system ("Pause");  
    return 0;  
}  
float promedio (float a, float b){  
    float prom;  
    a = a + 3;  
    b = b + 3;  
    prom = (a + b)/2;  
    return prom;  
}
```

En este ejercicio, nosotros ingresamos 2 valores de tipo “int” y se guardan en las variables “a” y “b”, después, creamos una función llamada “maximo”, donde, hacemos una condición para saber si “a” es mayor que “b”, si es verdadera, entonces, imprimirá el valor de “a”, pero si es falsa, imprimirá el valor de “b”, es lo que pide el programa.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int maximo (int a, int b);
5  int main(){
6      int a, b, c;
7      puts("INGRESA EL PRIMER VALOR");
8      scanf("%d", &a);
9      puts("INGRESA EL SEGUNDO VALOR");
10     scanf("%d", &b);
11     c = maximo(a, b);
12     printf("El valor mas grande: %d \n", c);
13
14     system ("Pause");
15     return 0;
16 }
17 int maximo (int a, int b){
18     int c;
19     if(a > b){
20         c = a;
21     }else{
22         c = b;
23     }
24     return c;
25 }
```

En este caso, el primer valor es 100 y el segundo es 99, tendría que arrojar el valor más grande y eso hace.

```
INGRESA EL PRIMER VALOR
100
INGRESA EL SEGUNDO VALOR
99
El valor mas grande: 100
Presione una tecla para continuar . . .
```

Este ejercicio nos pedía sacar el área de un triángulo, para ello, necesitábamos pedir 3 valores desde consola, “a”, “b” y “c”, creamos una función de tipo “float” llamada “triangulo”, con los argumentos también de tipo “float”, para sacar el área, tuvimos que sumar los 3 valores que ingresamos y dividir el resultado entre 2, eso se guarda en la variable “p”, después, a eso se le resta el valor de cada uno de los 3 valores para después multiplicarse, por último, se saca la raíz y se guarda en la variable “d” y se imprimirá en pantalla.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  float triangulo (float a, float b, float c);
5  int main(){
6      float a, b, c;
7      float d;
8      d = triangulo(a, b, c);
9      printf("El area del triangulo es de: %f \n", d);
10     system ("Pause");
11     return 0;
12 }
13 float triangulo (float a, float b, float c){
14     float d, p, q;
15     puts("INGRESA EL PRIMER VALOR");
16     scanf("%f", &a);
17     puts("INGRESA EL SEGUNDO VALOR");
18     scanf("%f", &b);
19     puts("INGRESA EL SEGUNDO VALOR");
20     scanf("%f", &c);
21     p = (a + b + c)/2;
22     q = p*(p - a)*(p - b)*(p - c);
23     d = sqrt(q);
24     return d;
25 }
```

En este caso, el primer valor es 10, el segundo es 18 y el tercero es 20, hará todo el procedimiento de operaciones e imprimirá el resultado, el área fue de 89.799774.

```
INGRESA EL PRIMER VALOR
10
INGRESA EL SEGUNDO VALOR
18
INGRESA EL SEGUNDO VALOR
20
El area del triangulo es de: 89.799774
Presione una tecla para continuar . . .
```

## ***FUNCIONES RECURSIVAS***

Las funciones recursivas lo que hacen es que entra a una función y va a entrar muchas veces hasta que la condición se cumpla, en este caso, la función se llama “potencia”, dice que va a regresar cuando el valor de “b” sea menor que 1, sino el valor de “b” ira disminuyendo hasta que sea el valor que quiere la condición.

```
int main(void){
    int x,y, max;

    x = 2;
    y = 3;

    max = potencia(x,y);
    printf("La potencia es: %d ", max);
    system("Pause");
    return 0;
}

int potencia (a, b){
    if (b < 1)
        return 1;
    return a * potencia (a, b-1);
}
```



## **ESTRUCTURAS**

Las estructuras son muy utiles para declarar variables dentro de una estructura en especifico y al momento de seleccionarla, ya sabras de que variable se esta hablando. Este ejemplo muestra como se crea una estructura, en este caso, se llama "CD", dentro tiene 2 variables de tipo "char", "titulo" con un tamaño de 100 y "artista" con un tamaño de 50, para identificarla dentro del metodo main, se le coloca un identificador, en este caso, es "cd1", por consola se ingresara el titulo y se guardara en la variable "titulo" de la estructura "CD" y lo imprimira en pantalla.

```
struct CD{  
    char titulo[100];  
    char artista[50];  
};  
  
int main(void)  
{  
    struct CD cd1;  
    printf("Ingresa el titulo\n");  
    gets (cd1.titulo);  
  
    printf ("%s\n", cd1.titulo);  
  
    system("Pause");  
    return 0;  
}
```

Este ejercicio nos pide ingresarle titulo, numero de canciones, año y precio de un CD desde consola, creamos la estructura "CD" y añadimos las variables de tipo "char" que nos pidio, despues, en el metodo main, pedimos los valores desde consola y los guardamos en sus respectivas variables, por ultimo, imprimimos los valores de cada variable.

```
2  #include <stdlib.h>
3
4  struct CD{
5      char titulo[100];
6      char artista[50];
7      char numCanciones[10];
8      char anio[10];
9      char precio[10];
10 };
11 int main(void){
12     struct CD cd1;
13     printf("Ingresa el titulo \n");
14     gets(cd1.titulo);
15     puts("Ingresa el numero de canciones");
16     gets(cd1.numCanciones);
17     puts("Ingresa el año");
18     gets(cd1.anio);
19     puts("Ingresa el precio");
20     gets(cd1.precio);
21     printf("Titulo: %s\n", cd1.titulo);
22     printf("Numero de canciones: %s\n", cd1.numCanciones);
23     printf("Año: %s\n", cd1.anio);
24     printf("Precio: $%s\n", cd1.precio);
25     system("Pause");
26     return 0;
27 }
```

En este caso, el titulo es "Yellow Submarine", el numero de canciones es 340, el año es 2010 y el precio de 1200 y asi es como hace la impresión de pantalla.

```
Ingresa el titulo
Yellow Submarine
Ingresa el numero de canciones
340
Ingresa el año
2010
Ingresa el precio
1200

Titulo: Yellow Submarine
Numero de canciones: 340
Año: 2010
Precio: $1200
Presione una tecla para continuar . . .
```

Este ejercicio nos pide que ingresemos el nombre, el sexo y el sueldo de un empleado desde consola y que los muestre en pantalla, para ello, creamos la estructura “empleado”, donde añadimos 3 variables de tipo “char”, “nombre” con un tamaño de 100, “sexo” con un tamaño de 50 y “sueldo” con un tamaño de 10, el identificador que le colocamos en el metodo main es “emp1”, pedimos los datos y los guardamos en sus respectivas variables.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct empleado{
5      char nombre[100];
6      char sexo[50];
7      char sueldo[10];
8  };
9  int main(void){
10     struct empleado emp1;
11     puts("Ingresa el nombre del empleado");
12     gets(emp1.nombre);
13     puts("Ingresa el sexo del empleado");
14     gets(emp1.sexo);
15     puts("Ingresa el sueldo del empleado");
16     gets(emp1.sueldo);
17     printf("\nNombre: %s\n", emp1.nombre);
18     printf("Sexo: %s\n", emp1.sexo);
19     printf("Sueldo: $%s\n", emp1.sueldo);
20     system("Pause");
21     return 0;
22 }
```

En este caso, el nombre es “Alexis Luz Pedraza”, el sexo es “Masculino” y el sueldo es “2500”, así es como saldría la impresión de pantalla.

```
Ingresa el nombre del empleado
Alexis Luz Pedraza
Ingresa el sexo del empleado
Masculino
Ingresa el sueldo del empleado
2500

Nombre: Alexis Luz Pedraza
Sexo: Masculino
Sueldo: $2500
Presione una tecla para continuar . . .
```

Este ejercicio nos pide que ingresemos el nombre, el pais, el deporte y el numero de medallas de un atleta desde consola y que los muestre en pantalla, para ello, creamos la estructura “atleta”, donde añadimos 4 variables de tipo “char”, “nombre” con un tamaño de 100, “pais” con un tamaño de 50, “deporte” con un tamaño de 50 y “numMedallas” con un tamaño de 10, el identificador que le colocamos en el metodo main es “at1”, pedimos los datos y los guardamos en sus respectivas variables.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct atleta{
5      char nombre[100];
6      char pais[50];
7      char deporte[50];
8      char numMedallas[10];
9  };
10 int main(void){
11     struct atleta at1;
12     puts("Ingresa el nombre del atleta");
13     gets(at1.nombre);
14     puts("Ingresa de que pais es el atleta");
15     gets(at1.pais);
16     puts("Ingresa el deporte que practica");
17     gets(at1.deporte);
18     puts("Ingresa el numero de medallas del atleta");
19     gets(at1.numMedallas);
20     printf("\nNombre: %s\n", at1.nombre);
21     printf("Pais: %s\n", at1.pais);
22     printf("Deporte: %s\n", at1.deporte);
23     printf("Numero de medallas: %s\n", at1.numMedallas);
24     system("Pause");
25     return 0;
26 }
```

En este caso, el nombre es “George Wilson”, el pais es “Alemania”, el deporte es “Natacion” y el numero de medallas es 5, asi es como saldria la impresión de pantalla.

```
Ingresa el nombre del atleta
George Wilson
Ingresa de que pais es el atleta
Alemania
Ingresa el deporte que practica
Natacion
Ingresa el numero de medallas del atleta
5

Nombre: George Wilson
Pais: Alemania
Deporte: Natacion
Numero de medallas: 5
Presione una tecla para continuar . . .
```

## ARREGLO DE ESTRUCTURAS

Este ejercicio nos pide que ingresemos el nombre, el sexo y el sueldo de 5 empleados desde consola y que muestre los datos del que tiene menos sueldo en pantalla, para ello, creamos la estructura “empleado”, donde añadimos 3 variables de tipo “char”, “nombre” con un tamaño de 50, “sexo” con un tamaño de 50 y “sueldo” con un tamaño de 50, el identificador que le colocamos en el metodo main es “emp1” con un tamaño de 5, ya que, en cada espacio van a estar los datos de cada empleado, pedimos los datos con un ciclo for y los guardamos en sus respectivas variables, despues, comparamos el sueldo del primer empleado con el del segundo, y asi sucesivamente, hasta que llegue a la conclusion de que ese trabajador tiene el menor sueldo de los otros 4.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  struct empleado{
5      char nada[50];
6      char nombre[50];
7      char sexo[50];
8      int sueldo;
9  };
10 int main(){
11     struct empleado emp1[5];
12     int i, cant;
13     for(i = 0; i < 5; i++){
14         puts("");
15         gets(emp1[i].nada);
16         puts("\nIngresa el nombre del empleado");
17         gets(emp1[i].nombre);
18         puts("Ingresa el sexo del empleado");
19         gets(emp1[i].sexo);
20         puts("Ingresa el sueldo del empleado");
21         scanf("%d", &cant);
22         emp1[i].sueldo = cant;
23     }
24     if(emp1[0].sueldo < emp1[1].sueldo){
25         if(emp1[0].sueldo < emp1[2].sueldo){
26             if(emp1[0].sueldo < emp1[3].sueldo){
```



```
27 if(emp1[0].sueldo < emp1[4].sueldo){
28     printf("\nNombre: %s\n", emp1[0].nombre);
29     printf("Sexo: %s\n", emp1[0].sexo);
30     printf("Sueldo: $%d\n", emp1[0].sueldo);
31 }else{
32     printf("\nNombre: %s\n", emp1[4].nombre);
33     printf("Sexo: %s\n", emp1[4].sexo);
34     printf("Sueldo: $%d\n", emp1[4].sueldo);
35 }
36 }else{
37     if(emp1[3].sueldo < emp1[4].sueldo){
38         printf("\nNombre: %s\n", emp1[3].nombre);
39         printf("Sexo: %s\n", emp1[3].sexo);
40         printf("Sueldo: $%d\n", emp1[3].sueldo);
41     }else{
42         printf("\nNombre: %s\n", emp1[4].nombre);
43         printf("Sexo: %s\n", emp1[4].sexo);
44         printf("Sueldo: $%d\n", emp1[4].sueldo);
45     }
46 }
47 }else{
48     if(emp1[2].sueldo < emp1[3].sueldo){
49         if(emp1[2].sueldo < emp1[4].sueldo){
50             printf("\nNombre: %s\n", emp1[2].nombre);
51             printf("Sexo: %s\n", emp1[2].sexo);
52             printf("Sueldo: $%d\n", emp1[2].sueldo);
```

```
53         }else{
54             printf("\nNombre: %s\n", emp1[4].nombre);
55             printf("Sexo: %s\n", emp1[4].sexo);
56             printf("Sueldo: $%d\n", emp1[4].sueldo);
57         }
58     }else{
59         if(emp1[3].sueldo < emp1[4].sueldo){
60             printf("\nNombre: %s\n", emp1[3].nombre);
61             printf("Sexo: %s\n", emp1[3].sexo);
62             printf("Sueldo: $%d\n", emp1[3].sueldo);
63         }else{
64             printf("\nNombre: %s\n", emp1[4].nombre);
65             printf("Sexo: %s\n", emp1[4].sexo);
66             printf("Sueldo: $%d\n", emp1[4].sueldo);
67         }
68     }
69 }
70 }else{
71     if(emp1[1].sueldo < emp1[2].sueldo){
72         if(emp1[1].sueldo < emp1[3].sueldo){
73             if(emp1[1].sueldo < emp1[4].sueldo){
74                 printf("\nNombre: %s\n", emp1[1].nombre);
75                 printf("Sexo: %s\n", emp1[1].sexo);
76                 printf("Sueldo: $%d\n", emp1[1].sueldo);
77             }else{
78                 printf("\nNombre: %s\n", emp1[4].nombre);
```



```
79         printf("Sexo: %s\n", emp1[4].sexo);
80         printf("Sueldo: $%d\n", emp1[4].sueldo);
81     }
82 }else{
83     if(emp1[3].sueldo < emp1[4].sueldo){
84         printf("\nNombre: %s\n", emp1[3].nombre);
85         printf("Sexo: %s\n", emp1[3].sexo);
86         printf("Sueldo: $%d\n", emp1[3].sueldo);
87     }else{
88         printf("\nNombre: %s\n", emp1[4].nombre);
89         printf("Sexo: %s\n", emp1[4].sexo);
90         printf("Sueldo: $%d\n", emp1[4].sueldo);
91     }
92 }
93 }else{
94     if(emp1[2].sueldo < emp1[3].sueldo){
95         if(emp1[2].sueldo < emp1[4].sueldo){
96             printf("\nNombre: %s\n", emp1[2].nombre);
97             printf("Sexo: %s\n", emp1[2].sexo);
98             printf("Sueldo: $%d\n", emp1[2].sueldo);
99         }else{
100             printf("\nNombre: %s\n", emp1[4].nombre);
101             printf("Sexo: %s\n", emp1[4].sexo);
102             printf("Sueldo: $%d\n", emp1[4].sueldo);
103         }
104     }else{
105         if(emp1[3].sueldo < emp1[4].sueldo){
106             printf("\nNombre: %s\n", emp1[3].nombre);
107             printf("Sexo: %s\n", emp1[3].sexo);
108             printf("Sueldo: $%d\n", emp1[3].sueldo);
109         }else{
110             printf("\nNombre: %s\n", emp1[4].nombre);
111             printf("Sexo: %s\n", emp1[4].sexo);
112             printf("Sueldo: $%d\n", emp1[4].sueldo);
113         }
114     }
115 }
116 }
117
118     system("Pause");
119     return 0;
120 }
```



Ingresamos los nombres, el sexo y el sueldo de cada empleado, en este caso, el que tiene el menor sueldo es “Hector”, sexo “Masculino” y su sueldo “1000”, ese es el que imprime en pantalla.

```
Ingresa el nombre del empleado
Alexis
Ingresa el sexo del empleado
Maculino
Ingresa el sueldo del empleado
1500
```

```
Ingresa el nombre del empleado
Fernanda
Ingresa el sexo del empleado
Femenino
Ingresa el sueldo del empleado
2000
```

```
Ingresa el nombre del empleado
Hector
Ingresa el sexo del empleado
Masculino
Ingresa el sueldo del empleado
1000
```

```
Ingresa el nombre del empleado
Maria
Ingresa el sexo del empleado
Femenino
Ingresa el sueldo del empleado
5000
```

```
Ingresa el nombre del empleado
Gabriel
Ingresa el sexo del empleado
Masculino
Ingresa el sueldo del empleado
3000
```

```
Nombre: Hector
Sexo: Masculino
Sueldo: $1000
Presione una tecla para continuar . . .
```



## ENVIO DE ESTRUCTURAS A FUNCIONES

Este ejercicio nos ayuda para ingresar diferentes valores desde cada funcion en una misma estructura, en esta caso, creamos 4 funciones, una multiplicara las 2 fracciones, otra las sumara, otra las restara y la otra las dividira, haciendo esto un poco mas practico y sin tanto enredadero. El numerador de la primera fraccion es 4, su denominador es 3 y el numerador de la segunda fraccion es 2 y su denominador es 6.

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



```
27     return 0;
28 }
29 int suma(struct fraccion suma1, struct fraccion suma2){
30     int den = suma1.den * suma2.den;
31     int num = (suma2.den * suma1.num) + (suma1.den * suma2.num);
32     puts("SUMA");
33     printf("%d / %d \n\n", num, den);
34     return 0;
35 }
36 int resta(struct fraccion resta1, struct fraccion resta2){
37     int den = resta1.den * resta2.den;
38     int num = (resta2.den * resta1.num) - (resta1.den * resta2.num);
39     puts("RESTA");
40     printf("%d / %d \n\n", num, den);
41     return 0;
42 }
43 int division(struct fraccion division1, struct fraccion division2){
44     int den = division1.den * division2.num;
45     int num = division1.num * division2.den;
46     puts("DIVISION");
47     printf("%d / %d \n", num, den);
48     return 0;
49 }
```

Estos serían los resultados que mandaría a pantalla, la multiplicación, la suma, la resta y la división de fracciones.

```
MULTIPLICACION
8 / 18

SUMA
30 / 18

RESTA
18 / 18

DIVISION
24 / 6
Presione una tecla para continuar . . .
```