

Nombre de la práctica	EJERCICIOS EN CLASE			No.	1
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:

LENGUAJE C

El siguiente ejemplo muestra cómo es que el "printf" hace la impresión de pantalla, puedes mandar solo un texto o también que imprima el valor de una variable. En este caso solo se mandan textos para ver cómo es que funciona.

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

int main (){
    printf ("Este es un texto \n");
    printf ("que fue impreso desde C");
    printf ("\n");
    printf ("Ya soy todo un programador");
    printf ("\n");
    system("Pause");
    return 0;
}
```

Este es otro ejemplo de cómo se hace uso del “printf”, aquí mandamos a imprimir el texto “BIENVENIDO ALEXIS :)”

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      printf("BIENVENIDO ALEXIS :);");
6      system("Pause");
7      return 0;
8  }
```

Aquí ahora se va a imprimir el número 3, la suma de 3 más 4, para esto se hace uso de “%d”, este es para imprimir un número entero y el “%f” para un numero decimal (flotante), va a tomar el valor que esta después de la coma consecutivamente.

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

int main (){
    //Quiero imprimir el número 3
    printf("%d \n", 3);

    //Quiero imprimir la suma de 3 más 4
    printf("%d \n", 3+4);

    printf("La suma de %d + %d es = %d\n", 3,4,3+4);

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

Aquí esta otro ejemplo como el anterior, se toma el 5 para el primer “%d”, después se toma el 8 para el segundo “%d”, y por último, se toma la suma de 5+8 para el ultimo “%d”, se hace lo mismo en el segundo ejemplo de abajo.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      printf("La suma de %d + %d es = %d \n", 5, 8, 5+8);
6
7      printf("La suma de %d + %d es = %d \n", 78787, 3259, 78787+3259);
8
9      system("Pause");
10     return 0;
11 }
```

VARIABLES C

Este ejemplo muestra como se le deben asignar variables para que ya cuando mandemos la impresión solo pongamos la variable y no todo el número. Por ejemplo, el valor de “PI”.

```
#include <stdio.h>

#define PI 3.1415926

int main() {
    printf("Pi vale %f", PI);
    return 0;
}
```

Este ejemplo muestra como asignarle valor a una variable para que al momento de la impresión solo coloquemos la variable y no el número, esto se ocupa cuando nosotros metemos datos desde consola, se guardan en una variable y se usan.

```
#include <stdio.h>

int main (){
    int unNumero = 3;
    int otroNumero = 4;

    //Quiero imprimir la suma de 3 mas 4
    printf("%d", unNumero + otroNumero);

    printf ("\n");
    system("Pause");
    return 0;
}
```

Aquí se va a crear una variable de tipo entero llamada "una_variable", a este se le va a asignar el valor de 3, este valor se guardará en la memoria RAM, después se toma ese valor para sumarle 5 y ahora el valor que tiene la variable es 8, en la siguiente línea se hace un incremento de la variable, ahora su valor es de 9. Por último, se imprime el valor de la variable con un "printf".

```
#include <stdio.h>

int main (){
    int una_variable;
    una_variable = 3;
    una_variable = una_variable + 5;
    una_variable ++;

    printf("%d", una_variable);

    printf ("\n");
    system("Pause");
    return 0;
}
```

Aquí se crean dos variables de tipo flotante llamadas "r" y "pi", a "r" se le asigna el valor de 5, a "pi" se le asigna el valor de 3.1416, estos valores se guardarán en la memoria RAM, después se crea otra variable de tipo flotante llamada "área" donde se guardará lo que salga de la operación que es "pi * r * r". Por último, se imprime lo que está en esa variable con "%f".

```
#include <stdio.h>

int main (){
    double r = 5;
    double pi = 3.1416;
    double area = pi * r * r;

    printf("%f", area);

    printf ("\n");
    system("Pause");
    return 0;
}
```

Este ejemplo es parecido, se crean dos variables de tipo flotante, “a” y “A”, se va a hacer la suma del valor que tengan esas variables y se imprimirá el resultado.

```
#include <stdio.h>

int main (){
    double a = 3.1;
    double A = 4.5;

    printf("%f", a+A);

    printf ("\n");
    system("Pause");
    return 0;
}
```

Este ejemplo muestra los segundos que hemos vivido, para ello se hace uso de 5 variables de tipo entero, “min” que vale 60, “hora” que vale 60, “dia” que vale 24, “anio” que vale 365 y “edad” donde pondrás tu edad, en este ejemplo coloque 19, se hace la operación “edad*anio*dia*hora*min” para saber los segundos vividos.

```
1  #include <stdio.h>
2
3  int main(){
4      int min = 60;
5      int hora = 60;
6      int dia = 24;
7      int anio = 365;
8      int edad = 19;
9
10     printf("%d", edad*anio*dia*hora*min);
11     printf("\n");
12
13     system("Pause");
14     return 0;
15 }
```

Aquí hay dos ejemplos, el primero es declarar 3 variables de tipo entero y asignarles valores, hacer la multiplicación de esas 3 variables y mostrar el resultado.

El segundo es declarar 2 variables de tipo flotante y que hiciera la división de los valores que fueron asignados, y que hiciera la impresión de pantalla.

```
1  #include <stdio.h>
2
3  int main(){
4      int x = 5;
5      int y = 10;
6      int z = 15;
7
8      double a = 3.5;
9      double b = 10.5;
10
11     printf("La multiplicacion es %d", x*y*z);
12     printf("\n");
13     printf("La division es %f", a/b);
14     printf("\n");
15
16     system("Pause");
17     return 0;
18 }
```

Aquí se crean dos variables, una de tipo entero "r" y una de tipo flotante "pi", a "r" se le asigna el valor de 5, ya que, ese es el radio del diámetro de 10, a "pi" se le asigna el valor de 3.1416, estos valores se guardarán en la memoria RAM, después se crea otra variable de tipo flotante llamada "resultado" donde se guardará lo que salga de la operación que es "pi * r * r". Por último, se imprime lo que está en esa variable con "%f".

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int r = 5;
6      double pi = 3.1416;
7      double resultado = pi*r*r;
8
9      printf("El area del circulo es de = %f \n", resultado);
10
11     system("Pause");
12     return 0;
13 }
```

OPERADOR INCREMENTO/DECREMENTO

En este ejemplo se hace uso del post-incremento y del post-decremento, para ello, se colocan “++” después de la variable, eso va a hacer que se incremente lo que tiene la variable y se coloca “--” para hacer el decremento de lo que tiene la variable.

```
int main (){
    int var = 1;
    int con = 1;

    // Es equivalente a var = var+1;
    printf ("%d",var++);
    printf ("%d",var);

    // Es equivalente a con = con-1;
    printf ("%d",con--);
    printf ("%d",con);

    system ("Pause");
    return 0;
} // Fin main
```

En este ejemplo se hace uso del pre-incremento y del pre-decremento, para ello, se coloca “++” antes de la variable, eso va a hacer que se incremente lo que tiene la variable y después la utilice, y se coloca “--” para hacer el decremento de lo que tiene la variable y después usarla.

```
int main (){
    int var = 1;
    int con = 1;

    // Es equivalente a var = var+1;
    printf ("%d",++var);
    printf ("%d",var);

    // Es equivalente a con = con-1;
    printf ("%d",--con);
    printf ("%d",con);

    system ("Pause");
    return 0;
} // Fin main
```

En este ejemplo se hace uso del post-incremento y del pre-decremento, en el primero, se crea una variable de tipo entero llamada “c” con valor de 5, va a imprimir el valor de esa variable y después

hará el incremento, la variable quedaría con valor de 6. En el segundo, se crea una variable de tipo entero llamada “d” con valor de 10, primero hará el decremento y después hará la impresión, la variable quedaría con valor de 9.

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

int main (){
    int c = 5;

    printf ("c : %d\n", c);
    printf ("c++ : %d\n", c++);
    printf ("c : %d\n", c);

    int d = 10;
    printf ("d : %d\n", d);
    printf ("d : %d\n", --d);
    printf ("d : %d\n", d);

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

En este ejemplo se crean 2 variables de tipo entero, “x” y “y”, “x” con valor de 5 y “y” con valor de 10, después se crea otra variable de tipo entero llamada “z”, ahí se guardará el resultado de la operación “++x * y--”. Primero se hará el pre-incremento de “x”, su valor ahora es de 6, y después, se hará el post-decremento de “y”, su valor ahora es de 9. Por último, se hace la operación de los nuevos valores que tienen las variables “x” y “y” y se imprimen todas esas variables.

```
int main (){
    int x = 5;
    int y = 10;
    int z = ++x * y--;

    printf ("x : %d\n", x);
    printf ("y : %d\n", y);
    printf ("z : %d\n", z);

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

PRECEDENCIA DE OPERADORES

Este ejemplo hace referencia a la jerarquía de operaciones, se crean 6 variables con diferentes valores y otra solo para guardar el resultado de la operación “ $p * r \% q + w / x - y$ ”, primeramente, se haría la división, después la multiplicación, después la suma, y por último, la resta, el resultado sería de -6.

```
int main (){
    int p = 5;
    int q = 1;
    int r = 2;
    int w = 3;
    int x = 9;
    int y = 6;
    int z;

    z = p * r % q + w / x - y;
    printf ("z : %d \n", z);

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

En este ejemplo también se ve la jerarquía de operaciones, en la operación “ $1+2*3$ ” primero se hace la multiplicación, y después, la suma; en la operación “ $(1+2) * 3$ ” se hace primero la suma porque siempre se hace primero lo que está entre los paréntesis, y después, se hace la multiplicación; y en el último, se hace la multiplicación porque también está dentro de los paréntesis, y después, se hace la suma.

```
int main (){
    printf (" 1 + 2 : %d\n", 1+2*3);
    printf (" 1 + 2 : %d\n", (1+2)*3);
    printf (" 1 + 2 : %d\n", 1+(2*3));

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



En este ejemplo nos pide hacer unas operaciones, se crean 5 variables de tipo flotante, se les asignan valores y después se hace la operación “a + b + c / d” y se imprime, el resultado es 0.9768.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      double a = 0.333333333333;
6      double b = 0.6;
7      double c = 0.033333333333;
8      double d = 0.766666666667;
9      double e;
10
11     e = a + b + c / d;
12
13     printf("%f \n", e);
14
15     system("Pause");
16     return 0;
17 }
```

En este ejemplo nos pide hacer unas operaciones, se crean 6 variables de tipo entero y una de tipo flotante, se les asignan valores, y después, se hacen operaciones diferentes para que de él resultado y lo imprima, el resultado es 9.28.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int a = 5;
6      int b = 2;
7      int c = 1;
8      int d = 1;
9      int e = 2;
10     int f = 2;
11     double g = 0.25;
12     double h, i, j, k, l, m;
13
14     h = f - g;
15     i = e / h;
16     j = d + i;
17     k = c / j;
18     l = b / k;
19     m = a + l;
20
21     printf("%f \n", m);
22
23     system("Pause");
24     return 0;
25 }
```

OPERADORES LÓGICOS Y DE RELACIÓN

En este ejemplo se va a mostrar cómo es que se utilizan los operadores lógicos y de relación, para "AND" se utiliza "&", para "OR" se utiliza "|" y para "XOR" se utiliza "^". En "AND": "1 && 1" es verdadero, "1 && 0" es falso, "0 && 1" es falso y "0 && 0" es falso. En "OR": "1 || 1" es verdadero, "1 || 0" es verdadero, "0 || 1" es verdadero y "0 || 0" es falso. En "XOR": "1 ^^ 1" es falso, "1 ^^ 0" es verdadero, "0 ^^ 1" es verdadero y "0 ^^ 0" es falso.

```
int main (){
    printf (" ***** AND *****\n");
    printf (" true && true : %d\n", (1 && 1));
    printf (" true && false : %d\n", (1 && 0));
    printf (" false && true : %d\n", (0 && 1));
    printf (" false && false : %d\n", (0 && 0));

    printf (" ***** OR *****\n");
    printf (" true || true : %d\n", (1 || 1));
    printf (" true || false : %d\n", (1 || 0));
    printf (" false || true : %d\n", (0 || 1));
    printf (" false || false : %d\n", (0 || 0));

    printf (" ***** XOR *****\n");
    printf (" true ^ true : %d\n", (1 ^ 1));
    printf (" true ^ false : %d\n", (1 ^ 0));
    printf (" false ^ true : %d\n", (0 ^ 1));
    printf (" false ^ false : %d\n", (0 ^ 0));

    system ("Pause");
    return 0;
} // Fin main
```

En este ejemplo otra vez se hace uso de los operadores lógicos y de relación, se crean 4 variables de tipo entero, se hacen las operaciones para ver si dan verdadero o falso, con los operadores que ya hemos visto.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int p = 1;
6      int q = 0;
7      int r = 1;
8      int t = 0;
9
10     printf(" %d\n", p & r);
11     printf(" %d\n", q | t);
12     printf(" %d\n", p & q || r & t);
13     printf(" %d\n", p ^ q ^ r ^ t);
14     printf(" %d\n", !q & !t);
15     printf(" %d\n", !!!p);
16
17     system("Pause");
18     return 0;
19 }
```

En este ejemplo se utilizan los operadores de comparación, el primero dice "3>5", como el 3 no es mayor que el 5, el resultado dará falso; el segundo dice "3<5", como el 3 es menor que el 5, el resultado será verdadero; el tercero dice "3==5", como el 3 no es igual al 5, el resultado será falso; el cuarto dice "3!=5", como el 3 es diferente al 5, el resultado será verdadero.

```
int main (){

    printf (" %d\n", 3>5);

    printf (" %d\n", 3<5);

    printf (" %d\n", 3==5);

    printf (" %d\n", 3!=5);

    system("Pause");
    return 0;
} // Fin main
```

En este ejemplo se crean 4 variables de tipo entero, se les asignan valores y se utilizan para compararse y ser utilizados con operadores lógicos, sus resultados serán falsos o verdaderos.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int w = 9;
6      int x = 3;
7      int y = 7;
8      int z = -2;
9
10     printf(" %d\n", x < y && w > z);
11     printf(" %d\n", x >= w ^ z == y);
12     printf(" %d\n", y <= x || x != w);
13     printf(" %d\n", w == 9 ^ x == 3);
14     printf(" %d\n", y > z && z < x);
15     printf(" %d\n", !w != 9);
16
17     system("Pause");
18     return 0;
19 }
```

ENTRADA Y SALIDA DE DATOS

Este es un ejemplo de cómo imprimir un mensaje simple en pantalla.

```
printf (" Esto es una impresión en pantalla \n");
```

Este es un ejemplo de cómo imprimir un mensaje, y a la vez, valores de variables.

```
printf (" El valor de x es %d ", x );
```

Aquí se muestra la diferencia de utilizar el “printf” y el “puts”, el “puts” es utilizado para mandar un mensaje con un salto de línea incluido al final y es exclusivo solo para textos, el “printf” no hace eso, tú tienes que colocarle el salto de línea al final con “\n”.

```
printf ("Hola mundo \n");  
  
puts ("Hola mundo");
```

El “scanf” se utiliza para que tu introduzcas un número y se le asignes a una variable definida, a la variable se lo coloca el “&” antes para decir que se coloque el valor introducido desde consola a esa variable.

```
scanf( "%d %f" , &x, &precio);
```

IF

En este ejemplo se utiliza el “if”, una estructura de condición simple, aquí si la condición es verdadera, ejecuta lo que está dentro de la condición.

```
int main(){  
  
    if (true){  
        System.out.println ("Todo este ");  
        System.out.println ("código, se imprime dentro ");  
        System.out.println ("del bloque if ");  
    }  
    return 0;  
  
} // Fin main
```

En este ejemplo se crea una variable de tipo entero llamada “numero” con valor de 34, después, hay una condición “numero %2 == 0”, si el residuo de 34 % 2 es 0, entonces va a entrar a la condición y va a imprimir que el número es par.

```
#include <stdio.h>  
  
int main (){  
    int numero = 34;  
    if (numero %2 == 0){  
        printf("El numero es par\n");  
    }  
    printf ("Fin del programa\n");  
    system ("Pause");  
    return 0;  
}
```


En este ejemplo se crea una variable de tipo entero llamada “numero” con valor de 55, después, hay una condición “numero %2 == 0”, si el residuo de 55 % 2 es 0, entonces va a entrar a la condición y va a imprimir que el número es par, pero en este caso, el residuo no da cero, así que, no entrara a la condición y el programa seguirá su curso.

```
#include <stdio.h>

int main (){
    int numero = 55;
    if (numero %2 == 0){
        printf("El numero es par\n");
    }
    printf ("Fin del programa\n");
    system ("Pause");
    return 0;
}
```

En este ejemplo se tiene que introducir una calificación desde consola, la cual se guardará en la variable de tipo entero “a”, si la calificación es mayor o igual a 8 entonces entrará a la condición y se imprimirá un mensaje a pantalla, pero sino entonces no entrara a la condición y el programa seguirá con lo siguiente.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int a;
6      puts("INGRESA TU CALIFICACION");
7      scanf("%d", &a);
8      if(a >= 8){
9          puts("¡FELICIDADES APROBASTE EL CURSO!");
10     }
11     system("Pause");
12     return 0;
13 }
```

En este ejemplo se hace uso de la estructura de decisión simple anidada, que consta de dos “if”. Se crea una variable de tipo entero “numero” con valor de 55, hay una condición que si el número es menor que 100 entonces entra a ejecutar lo que está dentro de esa condición, imprime un mensaje a pantalla, y después, hay otra condición que, si el número es mayor a 50, entonces entra a ejecutar lo que está dentro de esa condición.

```
#include <stdio.h>

int main (){
    int numero = 55;
    if (numero < 100){
        printf("El número es menor que 100
        if (numero > 50)
            printf(" y mayor que 50 \n");
        }
    printf ("Fin del programa\n");
    system ("Pause");
    return 0;
}
```

En este ejemplo ahora se cambia el valor de “numero”, que ahora es 135, la condición dice que, si número es menor que 100 entonces entra a ejecutar lo que está dentro, pero en esta ocasión no es verdadero así que no entra.

```
#include <stdio.h>

int main (){
    int numero = 135;
    if (numero < 100){
        printf("El número es menor que 100
        if (numero > 50)
            printf(" y mayor que 50 \n");
        }
    printf ("Fin del programa\n");
    system ("Pause");
    return 0;
}
```

En este ejemplo se busca saber si una persona es acreedora de un crédito hipotecario, la persona debe introducir el sueldo y los años de antigüedad que tiene en el trabajo, si la persona tiene 5 años o más entonces entra a la primera condición y si su sueldo es mayor a 1000 entonces entra a la segunda condición y es sujeto al crédito hipotecario.

```
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int sueldo, anio;
6      double x;
7      puts("INGRESA TU SUELDO MENSUAL");
8      scanf("%d", &sueldo);
9      puts("INGRESA AÑOS DE ANTIGÜEDAD");
10     scanf("%d", &anio);
11     if(anio >= 5){
12         x = (sueldo * 100)/10;
13         if(x > 1000){
14             puts("ERES SUJETO A UN CREDITO HIPOTECARIO");
15         }
16     }
17     system("Pause");
18     return 0;
19 }
```

En este ejemplo se hace uso de la estructura de decisión doble, se crea una variable de tipo entero "numero" con valor de 36, si número es menor que 50 entonces entra a la condición e imprime un mensaje, pero sino entonces imprime otro.

```
#include <stdio.h>

int main (){
    int numero = 36;
    if (numero < 50)
        printf("El número es menor que 50 ")
    else
        printf("El número es mayor que 50")
    printf ("Fin del programa\n");
    system ("Pause");
    return 0;
}
```