



<b>NOMBRE DE LA PRÁCTICA</b>	<b>INTRODUCCIÓN AL LENGUAJE C.</b>			<b>NO.</b>	<b>1</b>
<b>ASIGNATURA:</b>	<b>MÉTODOS NUMÉRICOS.</b>	<b>CARRERA:</b>	<b>INGENIERÍA EN SISTEMAS COMPUTACIONALES</b>	<b>DURACIÓN DE LA PRÁCTICA (HRS)</b>	<b>2 HORAS</b>

**NOMBRE DEL ALUMNO:** Cristian Miranda Padilla

**GRUPO:** 3401

### I. Competencia(s) específica(s):

- Aplica los tipos de errores para identificar la incertidumbre y limitaciones de los cálculos numéricos en una computadora.

Encuadre con CACEI: Registra el (los) atributo(s) de egreso y los criterios de desempeño que se evaluarán en esta práctica.

No. atributo	Atributos de egreso del PE que impactan en la asignatura
1	Reconocer la necesidad permanente de conocimiento adicional y tener la habilidad para localizar, evaluar, integrar y aplicar este conocimiento adecuadamente.

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

- Aula de clases.

### III. Material empleado:

- Computadora personal.
- SO Linux.
- Dev C++.

### IV. Desarrollo de la práctica:

## EJERCICIOS EN DEV C++ ELABORADOS EN CLASE

### Ejercicio 1 “Hola mundo”

En este primer programa codificamos nuestro primer “Hola Mundo” en C++, para lo cual aprendimos algunas reglas de sintaxis para hacer posible el correcto funcionamiento de nuestro programa.

```
Hello_Word.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main () {
5      printf("Hello world Cristian \n");
6      system("Pause");
7      return 0;
8  }
```

```
crm@crm-ThinkPad-L560: ~/Documentos/Metodos Numericos
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o ejecutable Hello_Word.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
ejecutable Hello_Word.cpp Hello_Word.exe
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./ejecutable
Hello world Cristian
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

### Ejercicio 2 “Segundo Texto”

En este programa codificamos nuestro segundo texto en C++, para lo cual aprendimos algunas reglas de sintaxis para hacer posible el correcto funcionamiento de nuestro programa.

```
sses Hello_Word.cpp Texto.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      printf("Este es un texto \n");
6      printf("que fue impreso desde C");
7      printf("\n");
8      printf("Ya soy todo un programador");
9      printf("\n");
10     system("Pause");
11     return 0;
12 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ nano Texto.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o Texto Texto.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
ejecutable Hello_Word.cpp Hello_Word.exe Texto Texto.cpp Texto.exe
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Texto
Este es un texto
que fue impreso desde C
Ya soy todo un programador
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```



## Ejercicio 3 “Primera Suma”

En este programa codificamos nuestra primera suma decimal en C++, para lo cual aprendimos algunas reglas de sintaxis numérica para hacer posible el funcionamiento de dicho programa.

```

Hello_Word.cpp | Texto.cpp | Suma1.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      //Quiero imprimir el numero 3;
6      printf("%d \n", 3);
7
8      //Quiero imprimir la suma de 3 mas 4;
9      printf("%d \n", 3+4);
10
11     printf("La suma de %d + %d es = %d\n", 3, 4, 3+4);
12     system("Pause");
13     return 0;
14 }

crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o Suma Suma1.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
ejecutable      Hello_Word.exe  Suma1.cpp      Texto         Texto.exe
Hello_Word.cpp  Suma           Suma1.exe      Texto.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Suma
3
7
La suma de 3 + 4 es = 7
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 4 “Operaciones Aritmeticas”

En este programa codificamos 2 operaciones aritméticas en C++, para lo cual aplicamos algunas reglas de sintaxis numérica para hacer posible el funcionamiento de dicho programa.

```

Hello_Word.cpp | Texto.cpp | Suma1.cpp | [*] Operaciones_Aritmeticas.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      //Quiero imprimir el numero 5;
6      printf("%d \n", 5);
7
8      //Quiero imprimir el numero 8;
9      printf("%d \n", 8);
10
11     printf("La suma de %d + %d es = %d\n", 5, 8, 5+8);
12     //Quiero imprimir el numero 78787;
13     printf("%d \n", 78787);
14
15     //Quiero imprimir el numero 3259;
16     printf("%d \n", 3259);
17
18     printf("La suma de %d + %d es = %d\n", 78787, 3259, 78787+3259);
19     system("Pause");
20     return 0;
21 }
22
```



```
crm@crm-ThinkPad-L560: ~/Documentos/Metodos Numericos
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o Operaciones Operaciones_Aritmeticas.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
Aritmeticas      Operaciones      Suma      Texto.cpp
ejecutable      'Operaciones Aritmeticas.cpp' Suma1.cpp Texto.exe
Hello_Word.cpp  'Operaciones Aritmeticas.exe' Suma1.exe
Hello_Word.exe  Operaciones_Aritmeticas.cpp  Texto
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Operaciones
5
8
La suma de 5 + 8 es = 13
78787
3259
La suma de 78787 + 3259 es = 82046
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 5 “Ejemplo declaración de variables”

En este programa codificamos 2 variables aritméticas con un valor definido en C++ de tipo entero, para lo cual aplicamos algunas reglas de sintaxis numérica para hacer posible el funcionamiento de dicho programa.

```
[*] Variables.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int unNumero = 3;
6      int otroNumero = 4;
7
8      //Quiero imprimir la suma de 3 mas 4
9      printf("El resultado de la suma es:");
10     printf(" es: ");
11     printf("%d", unNumero + otroNumero);
12
13     printf("\n");
14     system("Pause");
15     return 0;
16 }
17
```

```
crm@crm-ThinkPad-L560: ~/Documentos/Metodos Numericos
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o Variables Variables.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
Aritmeticas      'Operaciones Aritmeticas.cpp' Suma1.cpp  Variables
ejecutable      'Operaciones Aritmeticas.exe' Suma1.exe  Variables.cpp
Hello_Word.cpp  Operaciones_Aritmeticas.cpp  Texto     Variables.exe
Hello_Word.exe  Operaciones_Aritmeticas.exe  Texto.cpp
Operaciones     Suma                          Texto.exe
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Variables
El resultado de la suma es: es: 7
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```



## Ejercicio 6 "Segundos de vida"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, para lo cual aplicamos algunas reglas de sintaxis numérica y de operaciones, para obtener los segundos de vida.

```
Variables.cpp [Ejercicio_SV.cpp]
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int min = 60;
6     int hora = 60;
7     int dia = 24;
8     int ano = 365*20;
9
10    //Quiero imprimir la suma de 3 mas 4
11    printf("La cantidad de segundos que e vivido son:");
12
13    printf(" %d", ano*dia*hora*min );
14
15    printf("\n");
16    system("Pause");
17    return 0;
18 }
19
```

```
crm@crm-ThinkPad-L560: ~/Documentos/Metodos Numericos
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o Ejercicio_SV Ejercicio_SV.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ls
Aritmeticas      Hello_Word.exe      Suma      Variables
ejecutable       Operaciones         Suma1.cpp Variables.cpp
Ejercicio_SV     'Operaciones Aritmeticas.cpp' Suma1.exe Variables.exe
Ejercicio_SV.cpp 'Operaciones Aritmeticas.exe' Texto
Ejercicio_SV.exe Operaciones_Aritmeticas.cpp  Texto.cpp
Hello_Word.cpp   Operaciones_Aritmeticas.exe  Texto.exe
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Ejercicio_SV
La cantidad de segundos que e vivido son:630720000
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```





## Ejercicio 7 "Multiplicación"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, para lo cual aplicamos algunas reglas de sintaxis numérica y de operaciones, para obtener el resultado de nuestra multiplicación.

```
Multiplicacion_27.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main ()
5  {
6      int V_uno=3;
7      int V_dos=4;
8      int V_tres=6;
9
10     printf("El resultado de la multiplicacion es:");
11
12     printf(" %d", V_uno*V_dos*V_tres );
13
14     printf("\n");
15     system("Pause");
16     return 0;
17 }
```

```
crm@crm-ThinkPad-L560: ~/Documentos/Metodos Numericos
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Multiplicacion
El resultado de la multiplicacion es:72
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 8 "Operadores"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar.

```
Multiplicacion_27.cpp Operadores_ID.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int x=5;
7      int y=10;
8      int z=++x*y--;
9
10     printf("x : %d\n", x);
11     printf("y : %d\n", y);
12     printf("z : %d\n", z);
13
14     system("Pause");
15     return 0;
16 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Operadores
x : 6
y : 9
z : 60
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```



## Ejercicio 9 "Operadores"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar para resolver las tablas de verdad.

```
Multiplicacion_27.cpp Operadores_ID.cpp Operadores.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      printf("****AND****\n");
6      printf("true && true : %d\n", (1&&1));
7      printf("true && false : %d\n", (1&&0));
8      printf("false && true : %d\n", (0&&1));
9      printf("false && false : %d\n", (0&&0));
10     printf("****OR****\n");
11     printf("true || true : %d\n", (1||1));
12     printf("true || false : %d\n", (1||0));
13     printf("false || true : %d\n", (0||1));
14     printf("false || false : %d\n", (0||0));
15     printf("****XOR****\n");
16     printf("true ^ true : %d\n", (1^1));
17     printf("true ^ false : %d\n", (1^0));
18     printf("false ^ true : %d\n", (0^1));
19     printf("false ^ false : %d\n", (0^0));
20
21     system("Pause");
22     return 0;
23 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Operador
****AND****
true && true : 1
true && false : 0
false && true : 0
false && false : 0
****OR****
true || true : 1
true || false : 1
false || true : 1
false || false : 0
****XOR****
true ^ true : 0
true ^ false : 1
false ^ true : 1
false ^ false : 0
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```



## Ejercicio 10 "Operadores"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo entero, así como también algunos operadores aritméticos a usar para resolver las tablas de verdad.

```

Multiplicacion_27.cpp | Operadores_ID.cpp | Operadores.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main() {
5      printf("****AND****\n");
6      printf("true && true : %d\n", (1 & 1));
7      printf("true && false : %d\n", (1 & 0));
8      printf("false && true : %d\n", (0 & 1));
9      printf("false && false : %d\n", (0 & 0));
10     printf("****OR****\n");
11     printf("true || true : %d\n", (1 || 1));
12     printf("true || false : %d\n", (1 || 0));
13     printf("false || true : %d\n", (0 || 1));
14     printf("false || false : %d\n", (0 || 0));
15     printf("****XOR****\n");
16     printf("true ^ true : %d\n", (1 ^ 1));
17     printf("true ^ false : %d\n", (1 ^ 0));
18     printf("false ^ true : %d\n", (0 ^ 1));
19     printf("false ^ false : %d\n", (0 ^ 0));
20
21     system("Pause");
22     return 0;
23 }

```

```

crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ nano Operadores.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Operador
****AND****
true && true : 1
true && false : 0
false && true : 0
false && false : 0
****OR****
true || true : 1
true || false : 1
false || true : 1
false || false : 0
****XOR****
true ^ true : 0
true ^ false : 1
false ^ true : 1
false ^ false : 0
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$

```

## Ejercicio 11" Operadores Lógicos"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo double, así como también algunos operadores con el fin de resolver un problema con lógico planteado.

```

Fracciones.cpp | Area_Tarea.cpp | Fraccion2_2.cpp | [*] OperadorC.cpp
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("Enunciado uno: %d\n", ((x < y) & (w > z)));
11     printf("Enunciado 2: %d\n", ((x >= w) ^ (z = y)));
12     printf("Enunciado 3: %d\n", ((y <= x) ^ (x != w)));
13     printf("Enunciado 4: %d\n", ((w = 9) ^ (x = 3)));
14     printf("Enunciado 5: %d\n", ((y > z) & (z = x)));
15     printf("Enunciado 6: %d\n", (!w != 9));
16
17     system("Pause");
18     return 0;
19 }
20

```





```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./OperadorC
Enunciado uno: 1
Enunciado 2: 7
Enunciado 3: 1
Enunciado 4: 0
Enunciado 5: 0
Enunciado 6: 1
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 12" Operadores Lógicos"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo double, así como también algunos operadores con el fin de resolver un problema con lógico planteado.

```
Fracciones.cpp | Area_Tarea.cpp | Fraccion2_2.cpp | OperadorC.cpp | OperadorU.cpp
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int x=5;
7      int y=10;
8      int z=++x*y--;
9
10     printf("x : %d\n", x);
11     printf("y : %d\n", y);
12     printf("x : %d\n", z);
13
14     system("Pause");
15     return 0;
16 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./OperadorU
x : 6
y : 9
x : 60
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 13" Área del Triángulo"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo double, así como también algunos operadores aritméticos y funciones de seno y coseno a usar para resolver un problema con el teorema de pitagoras.

```
1  #include <stdio.h>
2  #include <math.h>
3  #include <stdlib.h>
4  #define PI 3.14159265
5
6  int main()
7  {
8      double hip=100;
9      double ang=45;
10     double cat_op;
11     double cat_ad;
12     double AT;
13     double AC;
14     cat_op=sin(ang*PI/180)*100;
15     printf("Dimension del cateto opuesto: %.3f\n", cat_op);
16     cat_ad=cos(ang*PI/180)*100;
17     printf("\n Dimension del cateto adyacente: %.3f\n", cat_ad);
18     AT=((cat_op*cat_ad)/2);
19     printf("\n El area del tringulo es: %.3f\n", AT);
20     AC=(AT/2);
21     printf("\n El area del cuadrado es: %.3f\n", AC);
22     system("Pause");
23 }
```



```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Area
Dimension del cateto opuesto: 70.711

Dimension del cateto adyacente: 70.711

El area del triangulo es: 2500.000

El area del cuadrado es: 1250.000
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 14 "Fracción 1"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo double, así como también algunos operadores aritméticos ordenados para resolver un problema de acuerdo a la prioridad de operadores.

```
Fracciones.cpp | Area_Tarea.cpp | Fraccion2_2.cpp | OperadorC.cpp |
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4
5      double a=1;
6      double b=3;
7      double c=5;
8      double d=30;
9      double e=23;
10
11     double z;
12
13     z=((a/b)+(b/c)+(a/d)/(e/d));
14     printf("Resultado: %f \n", z);
15
16     system("Pause");
17     return 0;
18 }
19
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Frac
Resultado: 0.976812
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 15 "Fracción 2"

En este programa codificamos nuestras variables aritméticas con un valor definido en C++ de tipo double, así como también algunos operadores aritméticos ordenados para resolver un problema de acuerdo a la prioridad de operadores.

```
Fracciones.cpp | Area_Tarea.cpp | Fraccion2_2.cpp | OperadorC.cpp |
1  #include <stdio.h>
2  #include <stdlib.h>
3  int main(){
4
5      double a=5;
6      double b=2;
7      double c=1;
8      double d=4;
9
10     double e;
11
12     e=(a+(b/(c/(c+(b/(b-(c/d)))))));
13     printf("Resultado: %f \n", e);
14
15     system("Pause");
16     return 0;
17 }
18
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./Fraccion
Resultado: 9.285714
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 16 "Numero par e impar"

En este programa codificamos una variable del tipo entero con un valor definido y posteriormente con el uso de la condicional "if" evaluamos si este es par.

```
[*] Untitled1 [*] EjercicioClase.cpp
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero = 55;
6     if (numero %2 == 0){
7         printf("El numero es par\n");
8     }
9     printf("Fin del programa\n");
10    system("Pause");
11    return 0;
12 }

crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase
El numero es impar
Fin del programa
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 17 "Evaluación y clasificación de un curso"

En este programa codificamos una variable con ingreso desde teclado que representa una calificación y posteriormente con el uso de la condicional "if" evaluamos si esta es óptima o no para aprobar un curso.

```
[*] Untitled1 [*] EjercicioClase.cpp [*] EjercicioClase2.cpp
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int numero;
6     printf("Ingresa la calificacion del curso: \n");
7     scanf("%d", &numero);
8     if (numero >=7){
9         printf("Felicidades aprobaste el curso!\n");
10    } else{
11        printf("La calificacion minima aprobatoria es: 8\n");
12    }
13    system("Pause");
14    return 0;
15 }
```



```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o EjercicioClase2 EjercicioClase2.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase2
Ingresa la calificación del curso:
8
Felicitades aprobaste el curso!
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase2
Ingresa la calificación del curso:
6
La calificación mínima aprobatoria es: 8
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```

## Ejercicio 18” Evaluación antigüedad”

En este programa codificamos dos variables con ingreso desde teclado que representa la antigüedad y sueldo de un empleo para adquirir un crédito hipotecario y posteriormente con el uso de la condicional “if” evaluamos si es aceptado o no el usuario.

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(){
5     int antigüedad;
6     int sueldo;
7     int x;
8     printf("Años de antigüedad:\n");
9     scanf("%d", &antigüedad);
10
11     if (antigüedad >= 5){
12         printf("Sueldo:\n");
13         scanf("%d", &sueldo);
14         x = (sueldo * 100) / 10;
15         if (x >= 1000){
16             printf("Aceptado, Felicitades!\n");
17         }
18     } else{
19         printf("No Aceptado!\n");
20     }
21 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ g++ -o EjercicioClase3 EjercicioClase3.cpp
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase3
Años de antigüedad:
2
No Aceptado
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase3
Años de antigüedad:
10
Sueldo:
1000
Aceptado, Felicitades
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$
```





## Ejercicio 19 "Horas de trabajo"

Realizar el cálculo el salario semanal de un empleado, sabiendo que éste se calcula en base a las horas semanales trabajadas y de acuerdo a un precio especificado por hora. Si se pasa de las 40 horas semanales, las horas extras se pagarán a razón de 1.5 veces la hora ordinaria.

```
[*] EjercicioClase4.cpp | EjercicioClase.cpp | EjercicioClase2.cpp | EjercicioClase3.cpp | Area_Tarea.cpp |
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main(){
5      int HST;
6      int PH;
7      int SS;
8      int SSE;
9
10     printf("Horas semanales trabajadas:\n");
11     scanf("%d", &HST);
12     printf("Pago por hora:\n");
13     scanf("%d", &PH);
14     SS=(HST * PH);
15     if (HST <=40){
16
17         printf("El sueldo semanal es: $ %d \n", SS);
18
19     }else if (HST >40){
20         SSE=(1.5 * (HST-40)*PH);
21         printf("El sueldo extra es: $ %d \n", SSE);
22         SS=(40 * PH + (PH * 1.5 * (HST-40)));
23         printf("sueldo TOTAL semanal con extra: $ %d \n", SS);
24     }
25     system("Pause");
26     return 0;
27 }
28
```

```
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase4
Horas semanales trabajadas:
10
Pago por hora:
10
El sueldo semanal es: $ 100
sh: 1: Pause: not found
crm@crm-ThinkPad-L560:~/Documentos/Metodos Numericos$ ./EjercicioClase4
Horas semanales trabajadas:
42
Pago por hora:
10
El sueldo extra es: $ 30
sueldo TOTAL semanal con extra: $ 430
sh: 1: Pause: not found
```





## Ejercicio 20 "Cuadrados y Cubos"

Imprimir los cuadrados y los cubos de los primeros quinientos números naturales mediante la implementación del ciclo "while" que facilitara la codificación del código.

```

Untitled1 | EjercicioClase10.cpp | EjercicioClase8.cpp | EjercicioClase9.cpp | Eje
1  #include <stdio.h>
2  #include <stdlib.h>
3
4  int main()
5  {
6      int numero = 0;
7      int x;
8      int y;
9
10     while (numero <= 500) {
11         printf("%d, ", numero);
12         x = numero * numero;
13         y = numero * numero * numero;
14         numero++;
15         printf("%d, ", x);
16         printf("%d, ", y);
17     }
18
19     system("Pause");
20     return 0;

```

```

C:\MetodosNumericos> ./EjercicioClase9
0,0,0,1,1,1,2,4,8,3,9,27,4,16,64,5,25,125,6,36,216,7,49,343,8,64,512,9,81,729,10,100,1000,11,121,1331,12,144,1728,13,169,2197,14,196,2744,15,2
25,3375,16,256,4096,17,289,4913,18,324,5832,19,361,6859,20,400,8000,21,441,9261,22,484,10648,23,529,12167,24,576,13824,25,625,15625,26,676,175
76,27,729,19683,28,784,21952,29,841,24389,30,900,27000,31,961,29791,32,1024,32768,33,1089,35937,34,1156,39304,35,1225,42875,36,1296,46656,37,1
369,50653,38,1444,54872,39,1521,59319,40,1600,64000,41,1681,68921,42,1764,74088,43,1849,79507,44,1936,85184,45,2025,91125,46,2116,97336,47,220
9,103823,48,2304,110592,49,2401,117649,50,2500,125000,51,2601,132651,52,2704,140608,53,2809,148877,54,2916,157464,55,3025,166375,56,3136,17561
6,57,3249,185193,58,3364,195112,59,3481,205379,60,3600,216000,61,3721,226981,62,3844,238328,63,3969,250047,64,4096,262144,65,4225,274625,66,43
56,287496,67,4489,300763,68,4624,314432,69,4761,328509,70,4900,343000,71,5041,357911,72,5184,373248,73,5329,389017,74,5476,405224,75,5625,4218
75,76,5776,438976,77,5929,456533,78,6084,474552,79,6241,493039,80,6400,512000,81,6561,531441,82,6724,551368,83,6889,571787,84,7056,592704,85,7
225,614125,86,7396,636056,87,7569,658503,88,7744,681472,89,7921,704969,90,8100,729000,91,8281,753571,92,8464,778688,93,8649,804357,94,8836,830
584,95,9025,857375,96,9216,884736,97,9409,912673,98,9604,941192,99,9801,970299,100,10000,1000000,101,10201,1030301,102,10404,1061208,103,10609
,1092727,104,10816,1124864,105,11025,1157625,106,11236,1191016,107,11449,1225043,108,11664,1259712,109,11881,1295029,110,12100,1331000,111,123
21,1367631,112,12544,1404928,113,12769,1442897,114,12996,1481544,115,13225,1520875,116,13456,1560896,117,13689,1601613,118,13924,1643032,119,1
4161,1685159,120,14400,1728000,121,14641,1771561,122,14884,1815848,123,15129,1860867,124,15376,1906624,125,15625,1953125,126,15876,2000376,127
,16129,2048383,128,16384,2097152,129,16641,2146689,130,16900,2197000,131,17161,2248091,132,17424,2299968,133,17689,2352637,134,17956,2406104,1
35,18225,2460375,136,18496,2515456,137,18769,2571353,138,19044,2628072,139,19321,2685619,140,19600,2744000,141,19881,2803221,142,20164,2863288
,143,20449,2924207,144,20736,2985984,145,21025,3048625,146,21316,3112136,147,21609,3176523,148,21904,3241792,149,22201,3307949,150,22500,33750
00,151,22801,3442951,152,23104,3511808,153,23409,3581577,154,23716,3652264,155,24025,3723875,156,24336,3796416,157,24649,3869893,158,24964,394
4312,159,25281,4019679,160,25600,4096000,161,25921,4173281,162,26244,4251528,163,26569,4330747,164,26896,4410944,165,27225,4492125,166,27556,4
574296,167,27889,4657463,168,28224,4741632,169,28561,4826809,170,28900,4913000,171,29241,5000211,172,29584,5088448,173,29929,5177717,174,30276
,5268024,175,30625,5359375,176,30976,5451776,177,31329,5545233,178,31684,5639752,179,32041,5735339,180,32400,5832000,181,32761,5929741,182,331
24,6028568,183,33489,6128487,184,33856,6229504,185,34225,6331625,186,34596,6434856,187,34969,6539203,188,35344,6644672,189,35721,6751269,190,3
6100,6859000,191,36481,6967871,192,36864,7077888,193,37249,7189057,194,37636,7301384,195,38025,7414875,196,38416,7529536,197,38809,7645373,198
,39204,7762392,199,39601,7880599,200,40000,8000000,201,40401,8120601,202,40804,8242408,203,41209,8365427,204,41616,8489664,205,42025,8615125,2
06,42436,8741816,207,42849,8869743,208,43264,8998912,209,43681,9129329,210,44100,9261000,211,44521,9393931,212,44944,9528128,213,45369,9663597
,214,45796,9800344,215,46225,9938375,216,46656,10077696,217,47089,10218313,218,47524,10360232,219,47961,10503459,220,48400,10648000,221,48841,
10793861,222,49284,10941048,223,49729,11089567,224,50176,11239424,225,50625,11390625,226,51076,11543176,227,51529,11697083,228,51984,11852352,
229,52441,12008989,230,52900,12167000,231,53361,12326391,232,53824,12487168,233,54289,12649337,234,54756,12812904,235,55225,12977875,236,55696,
13144256,237,56169,13312053,238,56644,13481272,239,57121,13651919,240,57600,13824000,241,58081,13997521,242,58564,14172488,243,59049,14348907
,244,59536,14526784,245,60025,14706125,246,60516,14886936,247,61009,15069223,248,61504,15252992,249,62001,15438249,250,62500,15625000,251,6300
1,15813251,252,63504,16003008,253,64009,16194277,254,64516,16387064,255,65025,16581375,256,65536,16777216,257,66049,16974593,258,66564,1717351
2,259,67081,17373979,260,67600,17576000,261,68121,17779581,262,68644,17984728,263,69169,18191447,264,69696,18399744,265,70225,18609625,266,707
56,18821096,267,71289,19034163,268,71824,19248832,269,72361,19465109,270,72900,19683000,271,73441,19902511,272,73984,20123648,273,74529,203464
17,274,75076,20570824,275,75625,20796875,276,76176,21024576,277,76729,21253933,278,77284,21484952,279,77841,21717639,280,78400,21952000,281,78
961,22188041,282,79524,22425768,283,80089,22665187,284,80656,22906304,285,81225,23149125,286,81796,23393656,287,82369,23639903,288,82944,239887
872,289,83521,24137569,290,84100,24389000,291,84681,24642171,292,85264,24897088,293,85849,25153757,294,86436,25412184,295,87025,25672375,296,8
7616,25934336,297,88209,26198073,298,88804,26463592,299,89401,26730899,300,90000,27000000,301,90601,27270901,302,91204,27543608,303,91809,2781

```



## Ejercicio 21 “Números divisibles”

Imprimir todos los números divisibles entre 3 mayores a cero y menores a mil usando la estructura del ciclo while.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int num = 3;
7     while (num < 1000) {
8         if (num % 3 == 0) {
9             cout << num << endl;
10        }
11        num += 3;
12    }
13    return 0;
14 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Pag26
3
6
9
12
15
18
21
24
27
30
33
36
39
42
45
969
972
975
978
981
984
987
990
993
996
999
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```

## Ejercicio 22 “Números divisibles\_2”

Imprimir todos los números que son divisibles entre 2 y entre 7, mayores a 0 y menores a mil, usando el ciclo “while”.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int num = 1;
7     while (num < 1000) {
8         if (num % 2 == 0 && num % 7 == 0) {
9             cout << num << endl;
10        }
11        num ++;
12    }
13    return 0;
14 }
```



```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Pag26_2
826
840
854
868
882
896
910
924
938
952
966
980
994
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```

## Ejercicio 23 “Enteros Positivos < 100”

Escribir todos los enteros positivos menores que 100 omitiendo aquellos que son divisibles por 7 usando el ciclo “while”.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int num = 1;
7     while (num < 100) {
8         if (num % 7 != 0) {
9             cout << num << endl;
10        }
11        num++;
12    }
13    return 0;
14 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Pag26_3
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
85
86
87
88
89
90
92
93
94
95
96
97
99
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```





## Ejercicio 24 “Arreglos”

Imprimir el arreglo de 10 elementos sin ayuda de ciclos.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int mi Arreglo[10];
7     mi Arreglo[1] = 10;
8     mi Arreglo[2] = 10;
9     mi Arreglo[3] = 10;
10    mi Arreglo[4] = 10;
11    mi Arreglo[5] = 10;
12    mi Arreglo[6] = 10;
13    mi Arreglo[7] = 10;
14    mi Arreglo[8] = 10;
15    mi Arreglo[9] = 10;
16    mi Arreglo[10] = 10;
17
18    cout << mi Arreglo[1] << endl;
19    cout << mi Arreglo[2] << endl;
20    cout << mi Arreglo[3] << endl;
21    cout << mi Arreglo[4] << endl;
22    cout << mi Arreglo[5] << endl;
23    cout << mi Arreglo[6] << endl;
24    cout << mi Arreglo[7] << endl;
25    cout << mi Arreglo[8] << endl;
26    cout << mi Arreglo[9] << endl;
27    cout << mi Arreglo[10] << endl;
28
29    return 0;
30 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_11
10
10
10
10
10
10
10
10
10
10
10
10
10
10
10
*** stack smashing detected ***: terminated
Abortado ('core' generado)
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```



## Ejercicio 25 “100 Posiciones”

Crear un arreglo de 100 posiciones y llenar el arreglo con la tabla del dos y mostrar el arreglo en pantalla utilizando la estructura de los arreglos.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int arreglo[100];
7     int i = 0;
8     while(i < 100) {
9         arreglo[i++] = (i) * 2;
10    }
11    i = 0;
12    while(i < 100) {
13        cout << "2 * " << (i + 1) << " = " << arreglo[i++] << endl;
14    }
15    return 0;
16 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_28
2 * 1= 0
2 * 2= 2
2 * 3= 4
2 * 4= 6
2 * 5= 8
2 * 6= 10
2 * 7= 12
2 * 8= 14
2 * 9= 16
2 * 10= 18
2 * 11= 20
2 * 12= 22
2 * 13= 24
2 * 14= 26
2 * 15= 28
2 * 16= 30
2 * 17= 32
2 * 18= 34
2 * 19= 36
2 * 20= 38
2 * 21= 40
2 * 22= 42
2 * 23= 44
2 * 24= 46
2 * 25= 48
2 * 26= 50
2 * 27= 52
2 * 28= 54
2 * 29= 56
2 * 30= 58
2 * 31= 60
2 * 32= 62
2 * 33= 64
2 * 34= 66
2 * 35= 68
```





## Ejercicio 26 “Arreglo Inverso”

Crear un arreglo de 100 posiciones y llenar el arreglo al orden inverso al índice y mostrar el arreglo en pantalla usando la jerarquía de los arreglos.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int vector[100];
6
7     // Llenar el arreglo en orden inverso
8     for (int i = 0; i < 100; i++) {
9         vector[i] = 100 - i;
10    }
11
12    // Mostrar el arreglo en pantalla
13    for (int i = 0; i < 100; i++) {
14        cout << "Vector[" << i << "] = " << vector[i] << endl;
15    }
16
17    return 0;
18 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_29
Vector[0] = 100
Vector[1] = 99
Vector[2] = 98
Vector[3] = 97
Vector[4] = 96
Vector[5] = 95
Vector[6] = 94
Vector[7] = 93
Vector[8] = 92
Vector[9] = 91
Vector[10] = 90
Vector[11] = 89
Vector[12] = 88
Vector[13] = 87
Vector[14] = 86
Vector[15] = 85
Vector[16] = 84
Vector[17] = 83
Vector[18] = 82
Vector[19] = 81
Vector[20] = 80
Vector[21] = 79
Vector[22] = 78
Vector[23] = 77
Vector[24] = 76
Vector[25] = 75
Vector[26] = 74
Vector[27] = 73
Vector[28] = 72
Vector[29] = 71
Vector[30] = 70
Vector[31] = 69
Vector[32] = 68
Vector[33] = 67
Vector[34] = 66
Vector[35] = 65
Vector[36] = 64
Vector[37] = 63
Vector[38] = 62
Vector[39] = 61
Vector[40] = 60
Vector[41] = 59
Vector[42] = 58
Vector[43] = 57
Vector[44] = 56
Vector[45] = 55
Vector[46] = 54
Vector[47] = 53
Vector[48] = 52
Vector[49] = 51
Vector[50] = 50
Vector[51] = 49
Vector[52] = 48
Vector[53] = 47
Vector[54] = 46
Vector[55] = 45
Vector[56] = 44
Vector[57] = 43
Vector[58] = 42
Vector[59] = 41
Vector[60] = 40
Vector[61] = 39
Vector[62] = 38
Vector[63] = 37
Vector[64] = 36
Vector[65] = 35
Vector[66] = 34
Vector[67] = 33
Vector[68] = 32
Vector[69] = 31
Vector[70] = 30
Vector[71] = 29
Vector[72] = 28
Vector[73] = 27
Vector[74] = 26
Vector[75] = 25
Vector[76] = 24
Vector[77] = 23
Vector[78] = 22
Vector[79] = 21
Vector[80] = 20
Vector[81] = 19
Vector[82] = 18
Vector[83] = 17
Vector[84] = 16
Vector[85] = 15
Vector[86] = 14
Vector[87] = 13
Vector[88] = 12
Vector[89] = 11
Vector[90] = 10
Vector[91] = 9
Vector[92] = 8
Vector[93] = 7
Vector[94] = 6
Vector[95] = 5
Vector[96] = 4
Vector[97] = 3
Vector[98] = 2
Vector[99] = 1
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```



## Ejercicio 27 “Operaciones Aritméticas”

Dados un par de arreglos realizar 5 operaciones aritméticas y obtener los resultados planteados.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int A[] = {3, 5, 6, 8, 4, 7, 8, 5, 3, 1};
6     int B[] = {3, 4, 6, 8, 9, 1, 2, 3, 0, 9};
7
8     // A[3] mod (B[2]/2)
9     int result1 = A[3] % (B[2]/2);
10    cout << "A[3] mod (B[2]/2) = " << result1 << endl;
11
12    // B[A[1]] - A[9]
13    int result2 = B[A[1]] - A[9];
14    cout << "B[A[1]] - A[9] = " << result2 << endl;
15
16    // A[0] + A[1+2]
17    int result3 = A[0] + A[1+2];
18    cout << "A[0] + A[1+2] = " << result3 << endl;
19
20    // A[5] + B[5]
21    int result4 = A[5] + B[5];
22    cout << "A[5] + B[5] = " << result4 << endl;
23
24    // (A[3]/B[2])/2
25    int result5 = (A[3]/B[2])/2;
26    cout << "(A[3]/B[2])/2 = " << result5 << endl;
27
28    return 0;
29 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_30
A[3] mod (B[2]/2) = 2
B[A[1]] - A[9] = 0
A[0] + A[1+2] = 11
A[5] + B[5] = 8
(A[3]/B[2])/2 = 0
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```

## Ejercicio 28 “Arreglo de 20 Posiciones”

Crea un arreglo 20 posiciones. Asígnale a cada elemento un valor. Calcula el promedio de todos los elementos y calcula la multiplicación de todos los elementos.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int arreglo[20] = {2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 1, 3, 5, 7, 9, 11, 13, 15, 17, 19};
6     int suma = 0;
7     int multiplicacion = 1;
8     float promedio;
9
10    // Calcular la suma de todos los elementos
11    for (int i = 0; i < 20; i++) {
12        suma += arreglo[i];
13    }
14
15    // Calcular el promedio
16    promedio = (float) suma / 20;
17
18    // Calcular la multiplicación de todos los elementos
19    for (int i = 0; i < 20; i++) {
20        multiplicacion *= arreglo[i];
21    }
22
23    cout << "El promedio de los elementos es: " << promedio << endl;
24    cout << "La multiplicación de los elementos es: " << multiplicacion << endl;
25
26    return 0;
27 }
```



```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_31
El promedio de los elementos es: 10.5
La multiplicación de los elementos es: -2102132736
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```

## Ejercicio 29 “Arreglo Suma”

Escriba un programa que reciba un número N del usuario y haga la suma de todos los números desde 1 hasta N, usando la estructura del ciclo FOR.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int numero, suma = 0;
8     int i;
9     cout << "ingrese el numero N";
10    cin >> numero;
11
12    for( i = 1; i <= numero; i++) {
13        cout << i << "+" << endl;
14        suma += i;
15    }
16
17    cout << "=" << suma << endl;
18
19    return 0;
20 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./For_26
ingrese el numero N6
1+
2+
3+
4+
5+
6+
= 21
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```



## Ejercicio 30 “Tabla de Valores”

Escriba un programa en C que utilice un ciclo para reproducir una tabla de valores especifica como la del ejemplo planteado.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int A = 3;
6
7     cout << "A\tA+2\tA+4\tA+6" << endl;
8
9     for (int i = 0; i < 5; i++) {
10         cout << A << "\t" << A+2 << "\t" << A+4 << "\t" << A+6 << endl;
11         A += 3;
12     }
13
14     return 0;
15 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./Arreglo_26_2
A      A+2      A+4      A+6
3       5       7       9
6       8      10      12
9      11      13      15
12     14     16     18
15     17     19     21
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```

## Ejercicio 31 “Escalinata de asteriscos”

Escribir un programa que reciba un número N del usuario e imprima una escalinata de N pisos de asteriscos ascendente y descendente.

```
1 #include <stdio.h>
2 int main() {
3     int a;
4     int b;
5     int c;
6     printf("(1 a N) Escribe el numero a llegar");
7     scanf("%d", &c);
8
9     for (int a=1; a<=c; a++) {
10         for (int b=1; b<=a; b++) {
11             printf("*");
12         }
13         printf("\n");
14     }
15     printf("\n");
16     for (int a=c; a>=1; a--) {
17         for (int b=a; b>=1; b--) {
18             printf("*");
19         }
20         printf("\n");
21     }
22     return 0;
23 }
```



```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./TrianguloDos
(1 a N) Escribe el numero a llegar5
*
**
***
****
*****
*****
****
***
**
*
```

## Ejercicio 32 “Tabla de Multiplicar”

Realizar un programa que muestre las tablas de multiplicar al ingresar un valor desde el teclado mediante el ciclo “FOR”.

```
1 #include <stdio.h>
2
3 int main () {
4     int a, b, c;
5     int d;
6
7     printf("Ejecute con cualquier numero: ");
8     scanf("%d", &d);
9
10    for(int a=1; a<=d; a++){
11
12        for(int b=1; b<=d; b++){
13            c=a*b;
14            printf("%d\t", c);
15        }
16        printf("\n");
17    }
18    return 0;
19 }
```

```
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$ ./TrianguloUno
Ejecute con cualquier numero: 10
2      2      3      4      5      6      7      8      9      10
2      4      6      8      10     12     14     16     18     20
3      6      9      12     15     18     21     24     27     30
4      8      12     16     20     24     28     32     36     40
5      10     15     20     25     30     35     40     45     50
6      12     18     24     30     36     42     48     54     60
7      14     21     28     35     42     49     56     63     70
8      16     24     32     40     48     56     64     72     80
9      18     27     36     45     54     63     72     81     90
10     20     30     40     50     60     70     80     90     100
crm@crm-ThinkPad-L560:~/Documentos/MetodosNumericos$
```





## V. Conclusiones

En esta unidad abarcamos temas muy interesantes, desde la manera elaborar un algoritmo en lo que fue un nuevo lenguaje de programación, hasta el aprender a manipular los distintos ciclos de programación que conocemos y del mismo modo abordamos el tema de arreglos, todos y cada uno de estos elementos de programación aunados a lo que es el manejo de números, en su mayoría los ejercicios son simples pero requieren de una buena capacidad de razonamiento, análisis y sobre todo de la capacidad para llevar los conocimientos de la teoría a la práctica, considero que con estos ejercicios me di cuenta de algunas habilidades de que necesito reforzar y así mismo temas que necesito repasar para poder comprenderlos en sus totalidad para llevarlos a la práctica.