

Programación Básica

Guía práctica

autor 1

M.Eng Ingeniería de Sistemas y Computación

autor 2

M.Eng Ingeniería de Sistemas y Computación

autor 3

M.Eng Ingeniería de Sistemas y Computación

26 de septiembre de 2018

Índice general

1. Variables y Operadores	1
1.1. Variables	1
1.2. Operadores	6
2. Estructura de Decisión	15
2.1. Estructura Si-No	15
2.2. Estructura de selección	24
3. Estructuras de Iteración	33
3.1. Iteración	33
4. Métodos	53
4.1. Métodos Iterativos	53
4.2. Métodos Recursivos	65
5. Estructuras de Datos	79
5.1. Arreglos	79
5.2. Matrices	89
6. Manejo de Archivos	105

Indice de Tablas

Indice de Gráficas

Indice de Programas

1.1. Lenguaje C \Leftarrow Ejercicio 1	1
1.2. Lenguaje Pyton \Leftarrow Ejercicio 1	1
1.3. Lenguaje C \Leftarrow Ejercicio 2	1
1.4. Lenguaje Pyton \Leftarrow Ejercicio 2	2
1.5. Lenguaje C \Leftarrow Ejercicio 3	2
1.6. Lenguaje Pyton \Leftarrow Ejercicio 3	2
1.7. Lenguaje C \Leftarrow Ejercicio 4	3
1.8. Lenguaje Pyton \Leftarrow Ejercicio 4	3
1.9. Lenguaje C \Leftarrow Ejercicio 5	3
1.10. Lenguaje Pyton \Leftarrow Ejercicio 5	4
1.11. Lenguaje C \Leftarrow Ejercicio 6	4
1.12. Lenguaje Python \Leftarrow Ejercicio 6	5
1.13. Lenguaje C \Leftarrow Ejercicio 7	5
1.14. Lenguaje Python \Leftarrow Ejercicio 7	5
1.15. Lenguaje C \Leftarrow Ejercicio 8	6
1.16. Lenguaje Python \Leftarrow Ejercicio 8	6
1.17. Lenguaje C \Leftarrow Ejercicio 9	6
1.18. Lenguaje Python \Leftarrow Ejercicio 9	7
1.19. Lenguaje C \Leftarrow Ejercicio 10	7
1.20. Lenguaje Python \Leftarrow Ejercicio 10	8
1.21. Lenguaje C \Leftarrow Ejercicio 11	8
1.22. Lenguaje Python \Leftarrow Ejercicio 11	8
1.23. Lenguaje C \Leftarrow Ejercicio 12	9
1.24. Lenguaje Python \Leftarrow Ejercicio 12	9
1.25. Lenguaje C \Leftarrow Ejercicio 13	9
1.26. Lenguaje Python \Leftarrow Ejercicio 13	10
1.27. Lenguaje C \Leftarrow Ejercicio 14	10
1.28. Lenguaje Python \Leftarrow Ejercicio 14	10
1.29. Lenguaje C \Leftarrow Ejercicio 15	11
1.30. Lenguaje Python \Leftarrow Ejercicio 15	11

1.31. Lenguaje C \Leftarrow Ejercicio 16	11
1.32. Lenguaje Python \Leftarrow Ejercicio 16	12
1.33. Lenguaje C \Leftarrow Ejercicio 17	12
1.34. Lenguaje Python \Leftarrow Ejercicio 17	12
1.35. Lenguaje C \Leftarrow Ejercicio 18	13
1.36. Lenguaje Python \Leftarrow Ejercicio 18	13
1.37. Lenguaje C \Leftarrow Ejercicio 19	13
1.38. Lenguaje Python \Leftarrow Ejercicio 19	14
2.1. Lenguaje C \Leftarrow Ejercicio 20	15
2.2. Lenguaje Python \Leftarrow Ejercicio 20	15
2.3. Lenguaje C \Leftarrow Ejercicio 21	16
2.4. Lenguaje Python \Leftarrow Ejercicio 21	16
2.5. Lenguaje C \Leftarrow Ejercicio 22	16
2.6. Lenguaje Python \Leftarrow Ejercicio 22	17
2.7. Lenguaje C \Leftarrow Ejercicio 23	18
2.8. Lenguaje Python \Leftarrow Ejercicio 23	18
2.9. Lenguaje C \Leftarrow Ejercicio 24	19
2.10. Lenguaje Python \Leftarrow Ejercicio 24	19
2.11. Lenguaje C \Leftarrow Ejercicio 25	20
2.12. Lenguaje Python \Leftarrow Ejercicio 25	20
2.13. Lenguaje C \Leftarrow Ejercicio 26	20
2.14. Lenguaje Python \Leftarrow Ejercicio 26	21
2.15. Lenguaje C \Leftarrow Ejercicio 27	21
2.16. Lenguaje Python \Leftarrow Ejercicio 27	22
2.17. Lenguaje C \Leftarrow Ejercicio 28	23
2.18. Lenguaje Python \Leftarrow Ejercicio 28	23
2.19. Lenguaje C \Leftarrow Ejercicio 29	24
2.20. Lenguaje Python \Leftarrow Ejercicio 29	24
2.21. Lenguaje C \Leftarrow Ejercicio 30	24
2.22. Lenguaje Python \Leftarrow Ejercicio 30	25
2.23. Lenguaje C \Leftarrow Ejercicio 31	26
2.24. Lenguaje Python \Leftarrow Ejercicio 31	27
2.25. Lenguaje C \Leftarrow Ejercicio 32	27
2.26. Lenguaje Python \Leftarrow Ejercicio 32	28
2.27. Lenguaje C \Leftarrow Ejercicio 33	28
2.28. Lenguaje Python \Leftarrow Ejercicio 33	29
2.29. Lenguaje C \Leftarrow Ejercicio 34	29
2.30. Lenguaje Python \Leftarrow Ejercicio 34	30
3.1. Lenguaje C \Leftarrow Ejercicio 35	33

3.2. Lenguaje Python \Leftarrow Ejercicio 35	33
3.3. Lenguaje C \Leftarrow Ejercicio 36	34
3.4. Lenguaje Python \Leftarrow Ejercicio 36	34
3.5. Lenguaje C \Leftarrow Ejercicio 37	34
3.6. Lenguaje Python \Leftarrow Ejercicio 37	35
3.7. Lenguaje C \Leftarrow Ejercicio 38	35
3.8. Lenguaje Python \Leftarrow Ejercicio 38	36
3.9. Lenguaje C \Leftarrow Ejercicio 39	36
3.10. Lenguaje Python \Leftarrow Ejercicio 39	36
3.11. Lenguaje C \Leftarrow Ejercicio 40	37
3.12. Lenguaje Python \Leftarrow Ejercicio 40	37
3.13. Lenguaje C \Leftarrow Ejercicio 41	37
3.14. Lenguaje Python \Leftarrow Ejercicio 41	38
3.15. Lenguaje C \Leftarrow Ejercicio 42	38
3.16. Lenguaje Python \Leftarrow Ejercicio 42	39
3.17. Lenguaje C \Leftarrow Ejercicio 43	39
3.18. Lenguaje Python \Leftarrow Ejercicio 43	40
3.19. Lenguaje C \Leftarrow Ejercicio 44	40
3.20. Lenguaje Python \Leftarrow Ejercicio 44	40
3.21. Lenguaje C \Leftarrow Ejercicio 45	41
3.22. Lenguaje Python \Leftarrow Ejercicio 45	41
3.23. Lenguaje C \Leftarrow Ejercicio 46	42
3.24. Lenguaje Python \Leftarrow Ejercicio 46	43
3.25. Lenguaje C \Leftarrow Ejercicio 47	43
3.26. Lenguaje Python \Leftarrow Ejercicio 47	44
3.27. Lenguaje C \Leftarrow Ejercicio 48	45
3.28. Lenguaje Python \Leftarrow Ejercicio 48	46
3.29. Lenguaje C \Leftarrow Ejercicio 49	46
3.30. Lenguaje Python \Leftarrow Ejercicio 49	47
3.31. Lenguaje C \Leftarrow Ejercicio 50	47
3.32. Lenguaje Python \Leftarrow Ejercicio 50	48
3.33. Lenguaje C \Leftarrow Ejercicio 51	48
3.34. Lenguaje Python \Leftarrow Ejercicio 51	48
3.35. Lenguaje C \Leftarrow Ejercicio 52	48
3.36. Lenguaje Python \Leftarrow Ejercicio 52	49
3.37. Lenguaje C \Leftarrow Ejercicio 53	50
3.38. Lenguaje Python \Leftarrow Ejercicio 53	50
3.39. Lenguaje C \Leftarrow Ejercicio 54	50
3.40. Lenguaje Python \Leftarrow Ejercicio 54	51

4.1. Lenguaje C \Leftarrow Ejercicio 55	53
4.2. Lenguaje Python \Leftarrow Ejercicio 55	53
4.3. Lenguaje C \Leftarrow Ejercicio 56	54
4.4. Lenguaje Python \Leftarrow Ejercicio 56	54
4.5. Lenguaje C \Leftarrow Ejercicio 57	55
4.6. Lenguaje Python \Leftarrow Ejercicio 57	55
4.7. Lenguaje C \Leftarrow Ejercicio 58	56
4.8. Lenguaje Python \Leftarrow Ejercicio 58	57
4.9. Lenguaje C \Leftarrow Ejercicio 59	57
4.10. Lenguaje Python \Leftarrow Ejercicio 59	58
4.11. Lenguaje C \Leftarrow Ejercicio 60	58
4.12. Lenguaje Python \Leftarrow Ejercicio 60	58
4.13. Lenguaje C \Leftarrow Ejercicio 61	59
4.14. Lenguaje Python \Leftarrow Ejercicio 61	59
4.15. Lenguaje C \Leftarrow Ejercicio 62	60
4.16. Lenguaje Python \Leftarrow Ejercicio 62	61
4.17. Lenguaje C \Leftarrow Ejercicio 63	62
4.18. Lenguaje Python \Leftarrow Ejercicio 63	64
4.19. Lenguaje C \Leftarrow Ejercicio 64	65
4.20. Lenguaje Python \Leftarrow Ejercicio 64	66
4.21. Lenguaje C \Leftarrow Ejercicio 65	66
4.22. Lenguaje Python \Leftarrow Ejercicio 65	67
4.23. Lenguaje C \Leftarrow Ejercicio 66	67
4.24. Lenguaje Python \Leftarrow Ejercicio 66	68
4.25. Lenguaje C \Leftarrow Ejercicio 67	68
4.26. Lenguaje Python \Leftarrow Ejercicio 67	68
4.27. Lenguaje C \Leftarrow Ejercicio 68	69
4.28. Lenguaje Python \Leftarrow Ejercicio 68	69
4.29. Lenguaje C \Leftarrow Ejercicio 69	70
4.30. Lenguaje Python \Leftarrow Ejercicio 69	70
4.31. Lenguaje C \Leftarrow Ejercicio 70	71
4.32. Lenguaje Python \Leftarrow Ejercicio 70	71
4.33. Lenguaje C \Leftarrow Ejercicio 71	72
4.34. Lenguaje Python \Leftarrow Ejercicio 71	72
4.35. Lenguaje C \Leftarrow Ejercicio 72	73
4.36. Lenguaje Python \Leftarrow Ejercicio 72	73
4.37. Lenguaje C \Leftarrow Ejercicio 73	74
4.38. Lenguaje Python \Leftarrow Ejercicio 73	74
4.39. Lenguaje C \Leftarrow Ejercicio 74	74

4.40. Lenguaje Python \Leftarrow Ejercicio 74	75
4.41. Lenguaje C \Leftarrow Ejercicio 75	75
4.42. Lenguaje Python \Leftarrow Ejercicio 75	76
4.43. Lenguaje C \Leftarrow Ejercicio 76	76
4.44. Lenguaje Python \Leftarrow Ejercicio 76	77
4.45. Lenguaje C \Leftarrow Ejercicio 77	77
4.46. Lenguaje Python \Leftarrow Ejercicio 77	78
5.1. Lenguaje C \Leftarrow Ejercicio 78	79
5.2. Lenguaje Python \Leftarrow Ejercicio 78	80
5.3. Lenguaje C \Leftarrow Ejercicio 79	80
5.4. Lenguaje Python \Leftarrow Ejercicio 79	82
5.5. Lenguaje C \Leftarrow Ejercicio 80	83
5.6. Lenguaje Python \Leftarrow Ejercicio 80	84
5.7. Lenguaje C \Leftarrow Ejercicio 81	85
5.8. Lenguaje Python \Leftarrow Ejercicio 81	85
5.9. Lenguaje C \Leftarrow Ejercicio 82	86
5.10. Lenguaje Python \Leftarrow Ejercicio 82	87
5.11. Lenguaje C \Leftarrow Ejercicio 83	88
5.12. Lenguaje Python \Leftarrow Ejercicio 83	89
5.13. Lenguaje C \Leftarrow Ejercicio 84	89
5.14. Lenguaje Python \Leftarrow Ejercicio 84	90
5.15. Lenguaje C \Leftarrow Ejercicio 85	91
5.16. Lenguaje Python \Leftarrow Ejercicio 85	92
5.17. Lenguaje C \Leftarrow Ejercicio 86	93
5.18. Lenguaje Python \Leftarrow Ejercicio 86	94
5.19. Lenguaje C \Leftarrow Ejercicio 87	94
5.20. Lenguaje Python \Leftarrow Ejercicio 87	95
5.21. Lenguaje C \Leftarrow Ejercicio 88	96
5.22. Lenguaje Python \Leftarrow Ejercicio 88	97
5.23. Lenguaje C \Leftarrow Ejercicio 89	98
5.24. Lenguaje Python \Leftarrow Ejercicio 89	100
5.25. Lenguaje C \Leftarrow Ejercicio 90	101
5.26. Lenguaje Python \Leftarrow Ejercicio 90	102
6.1. Lenguaje Python \Leftarrow Ejercicio 91	106
6.2. Lenguaje Python \Leftarrow Ejercicio 92	106
6.3. Lenguaje Python \Leftarrow Ejercicio 93	107
6.4. Lenguaje Python \Leftarrow Ejercicio 94	108
6.5. Lenguaje Python \Leftarrow Ejercicio 95	108
6.6. Lenguaje Python \Leftarrow Ejercicio 96	110

6.7. Lenguaje Python \Leftarrow Ejercicio 97	111
--	-----

Capítulo 1

Variables y Operadores

1.1. Variables

Ejercicio 1. Analizar, diseñar e implementar un programa de computador que pregunte al usuario el nombre, e imprima en pantalla un saludo de bienvenida acompañado del nombre que fue ingresado.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     string nombre;
6     cout << "Buen día, ingrese su nombre:" << endl;
7     cin >> nombre;
8     cout << "Bienvenido " << nombre << endl;
9     return 0;
10 }
```

Programa 1.1: Lenguaje C \Leftarrow Ejercicio 1

```
1 nombre = str(input("Buen día, ingrese su nombre: "))
2 print("Bienvenido " + str(nombre))
```

Programa 1.2: Lenguaje Python \Leftarrow Ejercicio 1

Ejercicio 2. Escribir un programa que calcule la hipotenusa de un triángulo rectángulo capturando el valor de los catetos (opuesto y adyacente).

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
```

```
4
5 int main(int argc, char *argv[]) {
6     int opuesto, adyacente;
7     float hipotenusa;
8     cout << "Ingrese el cateto opuesto ";
9     cin >> opuesto;
10    cout << "Ingrese el cateto adyacente ";
11    cin >> adyacente;
12    hipotenusa = sqrt(pow(opuesto, 2.0)+pow(adyacente, 2.0));
13    cout << "el valor de la hipotenusa es " << hipotenusa << endl;
14    return 0;
15 }
```

Programa 1.3: Lenguaje C \Leftarrow Ejercicio 2

```
1 from math import *
2
3 opuesto = int(input("ingrese el valor del cateto opuesto "))
4 adyacente = int(input("ingrese el valor del cateto adyacente "))
5 hipotenusa = sqrt(opuesto**2 + adyacente**2)
6 print("hipotenusa = " + str(hipotenusa))
```

Programa 1.4: Lenguaje Python \Leftarrow Ejercicio 2

Ejercicio 3. Escribir un programa que calcule el número de horas, minutos y segundos que hay en un valor x de segundos indicados por el usuario.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char *argv[]) {
5     int seg, min, hor;
6     cout << "ingrese el valor de los segundos ";
7     cin >> seg;
8     hor = seg / 3600;
9     seg = seg - hor * 3600;
10    min = seg / 60;
11    seg = seg - min * 60;
12    cout << "horas = " << hor << endl;
13    cout << "min = " << min << endl;
14    cout << "seg = " << seg << endl;
15    return 0;
16 }
```

Programa 1.5: Lenguaje C \Leftarrow Ejercicio 3

```
1 segundos = int(input("ingrese el numero los segundos "))
2 horas = int(segundos / 3600)
3 segundos = segundos - (horas * 3600)
4 minutos = int(segundos / 60)
5 segundos = segundos - (minutos * 60)
6 print("Horas = " + str(horas))
```

```
7 print("Minutos = " + str(minutos))
8 print("Segundos = " + str(segundos))
```

Programa 1.6: Lenguaje Python \Leftarrow Ejercicio 3

Ejercicio 4. Escribir un programa que calcule el equivalente en pies de una longitud de x metros sabiendo que 1 metro equivale a 39.27 pulgadas y 12 pulgadas equivalen a 1 pie.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     double metros;
6     double pies;
7     double pulgadas;
8
9     cout<<"Ingrese la cantidad de metros: "<<endl;
10    cin>>metros;
11    pulgadas=metros*39.27;
12    cout<<metros<<" metros equivalen a "<<pulgadas<<" pulgadas."<<endl;
13    pies=pulgadas/12;
14    cout<<metros<<" metros equivalen a "<<pies<<" pies."<<endl;
15
16    return 0;
17 }
```

Programa 1.7: Lenguaje C \Leftarrow Ejercicio 4

```
1 from math import *
2
3 metros = float(input("ingrese la cantidad de metros "))
4 pulgadas = metros * 39.27
5 print(str(metros) + " metros equivalen a " + str(pulgadas) + "
    pulgadas. ")
6 pies = pulgadas / 12
7 print(str(metros) + " metros equivalen a " + str(pies) + " pies. ")
```

Programa 1.8: Lenguaje Python \Leftarrow Ejercicio 4

Ejercicio 5. Escribir un programa que calcule la distancia entre dos puntos a partir de sus coordenadas (x1, y1), (x2, y2).

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4
5 int main() {
6
7     double d;
8     double x1;
9     double x2;
10    double y1;
```

```
11 double y2;
12
13 cout<<"Ingrese la coordenada x1 del punto 1:? ";
14 cin>>x1;
15 cout<<"Ingrese la coordenada y1 del punto 1:? ";
16 cin>>y1;
17 cout<<"Ingrese la coordenada x2 del punto 2:? ";
18 cin>>x2;
19 cout<<"Ingrese la coordenada y2 del punto 2:? ";
20 cin>>y2;
21 d=sqrt((x2-x1)*(x2-x1)+(y2-y1)*(y2-y1));
22 cout<<"La distancia entre ("<<x1<<","<<y1<<) y ("<<x2<<","<<y2<<)
    es "<<d<<endl;
23
24 return 0;
25 }
```

Programa 1.9: Lenguaje C \Leftarrow Ejercicio 5

```
1 from math import *
2
3 x1 = float(input("ingrese la coordenada x1 del punto 1:? "))
4 y1 = float(input("ingrese la coordenada y1 del punto 1:? "))
5 x2 = float(input("ingrese la coordenada x2 del punto 2:? "))
6 y2 = float(input("ingrese la coordenada y2 del punto 2:? "))
7 d = sqrt((x2 - x1) ** 2 + (y2 - y1) ** 2 )
8 print("La distancia entre (" + str(x1) + "," + str(y1) + ") y (" + str
    (x2) + "," + str(y2) + ") es" + str(d))
```

Programa 1.10: Lenguaje Python \Leftarrow Ejercicio 5

Ejercicio 6. Escribir un programa que lea dos números enteros positivos mayores que 1, A y B, y obtenga los valores de la división entera y el residuo de A dividido B.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     int a;
7     int b;
8
9     cout<<"Ingrese el valor de A (A > 1):? ";
10    cin>>a;
11    cout<<"Ingrese el valor de B (B > 1):? ";
12    cin>>b;
13    if (a>1 && b>1) {
14        cout<<"División entera "<<a<<"/"<<b<<" = "<<(a/b)<<endl;
15        cout<<"Residuo División "<<a<<"% "<<b<<" = "<<a%b<<endl;
16    } else {
17        cout<<"Error en el ingreso de los datos "<<a<<","<<b<<endl;
18    }
19
20    return 0;
}
```

21 }

Programa 1.11: Lenguaje C \Leftarrow Ejercicio 6

```
1 from math import *
2
3 A = int(input("Ingrese el valor de A (A > 1):? "))
4 B = int(input("Ingrese el valor de B (B > 1):? "))
5 if A > 1 and B > 1:
6     print("División entera " + str(A) + "/" + str(B) + " = " + str(A//B)
7         )
8     print("Residuo División " + str(A) + "%" + str(B) + " = " + str(A
9         %B))
10 else:
11     print("Error en el ingreso de los datos " + str(A) + "," + str(B))
```

Programa 1.12: Lenguaje Python \Leftarrow Ejercicio 6

Ejercicio 7. Escribir un programa que calcule la velocidad de un proyectil que recorre x Km en t minutos, expresando el resultado en metros/segundo (Velocidad = espacio/tiempo)

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     double distancia;
6     double tiempo;
7     double velocidad;
8
9     cout<<"Ingresa la distancia recorrida (Kilómetros): ";
10    cin>>distancia;
11    cout<<"Ingrese el tiempo del recorrido (Minutos): ";
12    cin>>tiempo;
13    distancia=distancia*1000;
14    tiempo=tiempo*60;
15    velocidad=distancia/tiempo;
16    cout<<"La velocidad del proyectil es "<<velocidad<<" m/s"<<endl;
17
18    return 0;
19 }
```

Programa 1.13: Lenguaje C \Leftarrow Ejercicio 7

```
1 distancia = float(input("Ingrese la distancia recorrida (Kilómetros):
2 "))
3 tiempo = float(input("Ingrese el tiempo del recorrido (Minutos): "))
4
5 distancia = distancia * 1000
6 tiempo = tiempo * 60
7 velocidad = distancia / tiempo
8 print("La velocidad del proyectil es " + str(velocidad) + " m/s")
```

Programa 1.14: Lenguaje Python \Leftarrow Ejercicio 7

Ejercicio 8. Escribir un programa que lea dos números enteros A y B, y obtenga los valores de la división entera de A dividido B y el residuo de esta sin hacer uso del operador módulo.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int a;
8     int b;
9     int cociente;
10    int residuo;
11
12    cout<<"Dividendo: ";
13    cin>>a;
14    cout<<"Divisor: ";
15    cin>>b;
16    cociente = a/b;
17    residuo = a - cociente * b;
18    cout<<"El cociente de "<<a<<"/"<<b<<" es "<<cociente<<endl;
19    cout<<"El residuo de "<<a<<"/"<<b<<" es "<<residuo<<endl;
20
21    return 0;
22 }
```

Programa 1.15: Lenguaje C \Leftarrow Ejercicio 8

```
1 a = int(input("Dividendo: "))
2 b = int(input("Divisor: "))
3 cociente = a // b
4 residuo = a - cociente * b
5 print("El cociente de " + str(a) + "/" + str(b) + " es " + str(
    cociente))
6 print("El residuo de " + str(a) + "/" + str(b) + " es " + str(residuo)
    )
```

Programa 1.16: Lenguaje Python \Leftarrow Ejercicio 8

1.2. Operadores

Ejercicio 9. Escribir un programa que muestre el resultado de sumar, restar, multiplicar y dividir dos números enteros mayores que cero leídos desde el teclado.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
```

```
6
7  int a;
8  int b;
9
10 cout<<"Ingrese el primer número: ";
11 cin>>a;
12 cout<<"Ingrese el segundo número: ";
13 cin>>b;
14 if (a>0 && b>0) {
15     cout<<"la suma de "<<a<<" + "<<b<<" es "<<a+b<<endl;
16     cout<<"la resta de "<<a<<" - "<<b<<" es "<<a-b<<endl;
17     cout<<"el producto de "<<a<<" * "<<b<<" es "<<a*b<<endl;
18     cout<<"la division de "<<a<<" / "<<b<<" es "<<a/b<<endl;
19 } else {
20     cout<<"Error en el ingreso de los datos "<<a<<" o "<<b<<endl;
21 }
22
23 return 0;
24 }
```

Programa 1.17: Lenguaje C \Leftarrow Ejercicio 9

```
1 from math import *
2
3 a = int(input("Ingrese el primer número: "))
4 b = int(input("Ingrese el segundo número: "))
5 if a > 0 and b > 0:
6     print("la suma de " + str(a) + " + " + str(b) + " es " + str(a+b))
7     print("la resta de " + str(a) + " - " + str(b) + " es " + str(a-b))
8     print("el producto de " + str(a) + " * " + str(b) + " es " + str(a
9         *b))
9     print("la división de " + str(a) + " / " + str(b) + " es " + str(a
10         //b))
10 else:
11     print("Error en el ingreso de los datos " + a + " o " + b)
```

Programa 1.18: Lenguaje Python \Leftarrow Ejercicio 9

Ejercicio 10. Escribir un programa que lea dos números enteros, los almacene en dos variables e intercambie su contenido.

```
1 // Intercambia el contenido de las variables n y m aplicando el
  operador XOR
2
3 #include<iostream>
4 #include<cmath>
5 using namespace std;
6
7 int main() {
8     int m;
9     int n;
10    cout<<"Variable 1? "<<endl;
11    cin>>n;
```

```
12 cout<<"Variable 2?"<<endl;
13 cin>>m;
14 cout<<"Antes del cambio, n = "<<n<<" m = "<<m<<endl;
15 n=n^m;
16 m=n^m;
17 n=n^m;
18 cout<<"Después del cambio, n = "<<n<<" m = "<<m<<endl;
19 return 0;
20 }
```

Programa 1.19: Lenguaje C \Leftarrow Ejercicio 10

```
1 from math import *
2
3 n = int(input("Variable 1? "))
4 m = int(input("Variable 2? "))
5 print("Antes del cambio, n= " + str(n) + " m= " + str(m))
6 n=n^m
7 m=n^m
8 n=n^m
9 print("Después del cambio, n= " + str(n) + " m= " + str(m))
```

Programa 1.20: Lenguaje Python \Leftarrow Ejercicio 10

Ejercicio 11. Escribir un programa que, dado un número total de horas, devuelve el número de semanas, días y horas equivalentes.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char **argv)
5 {
6     int horas, dias, semanas;
7     cout << "ingrese numero de horas" << endl;
8     cin >> horas;
9
10    semanas = horas / 168;
11    horas = horas % 168;
12    dias = horas / 24;
13    horas = horas % 24;
14
15    cout << semanas << " semanas " << dias << " dias " << horas << "
    horas " << endl;
16
17    return 0;
18 }
```

Programa 1.21: Lenguaje C \Leftarrow Ejercicio 11

```
1 horas = int(input("ingrese el numero de horas"))
2 semanas = horas / 168
3 horas = horas % 168
4
5 dias = horas / 24
```

```
6 horas = horas % 24
7
8 print "%i semanas    %i dias    %i horas" % (semanas, dias, horas)
```

Programa 1.22: Lenguaje Python \Leftarrow Ejercicio 11

Ejercicio 12. Escribir un programa que dado un número muestre en pantalla el doble del número y la mitad del número sin usar los operadores de división ni multiplicación

```
1 #include <iostream>
2 using namespace std;
3
4 int main(int argc, char **argv)
5 {
6     int numero, mitad, doble;
7     cout << "ingrese numero deseado" << endl;
8     cin >> numero;
9
10    doble = numero << 1;
11    mitad = numero >> 1;
12
13    cout << " numero = " << numero << " mitad = " << mitad << " doble = "
14           << doble << endl;
15    return 0;
16 }
```

Programa 1.23: Lenguaje C \Leftarrow Ejercicio 12

```
1 numero = int(input("ingrese el numero deseado"))
2 doble = numero << 1
3 mitad = numero >> 1
4
5 print "numero = %i  mitad = %i  doble = %i" % (numero, mitad, doble
6 )
```

Programa 1.24: Lenguaje Python \Leftarrow Ejercicio 12

Ejercicio 13. Escribir un programa que genere un número aleatorio entre 1 y 99 y muestre el resto de la división entera por 2, 3, 5 y 7

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4
5 using namespace std;
6
7 int main(int argc, char **argv)
8 {
9     int numero;
10    srand(time(0));
11
```

```
12 numero = 1 + rand() % 99;
13 cout << numero << " / " << 2 << " sobran " << numero % 2 << endl;
14 cout << numero << " / " << 3 << " sobran " << numero % 3 << endl;
15 cout << numero << " / " << 5 << " sobran " << numero % 5 << endl;
16 cout << numero << " / " << 7 << " sobran " << numero % 7 << endl;
17
18 return 0;
19 }
```

Programa 1.25: Lenguaje C \Leftarrow Ejercicio 13

```
1 from random import *
2
3 numero = randint(1,99)
4
5 print " %i / %i sobran %i" % (numero, 2, numero % 2)
6 print " %i / %i sobran %i" % (numero, 3, numero % 3)
7 print " %i / %i sobran %i" % (numero, 5, numero % 5)
8 print " %i / %i sobran %i" % (numero, 7, numero % 7)
```

Programa 1.26: Lenguaje Python \Leftarrow Ejercicio 13

Ejercicio 14. Escribir un programa que lea dos números, calcule la potencia del primero elevado al segundo y muestre el resultado en pantalla

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     int base;
9     int exponente;
10    double potencia;
11
12    cout << "ingrese el valor de la base: ";
13    cin >> base;
14
15    cout << "ingrese el valor del exponente: ";
16    cin >> exponente;
17
18    potencia = pow(base, exponente);
19
20    cout << base << " ^ " << exponente << " = " << potencia << endl;
21
22    return 0;
23 }
```

Programa 1.27: Lenguaje C \Leftarrow Ejercicio 14

```
1 base = int(input("ingrese el valor de la base: "))
2 exponente = int(input("ingrese el valor del exponente: "))
3
```

```
4 potencia = base ** exponente
5
6 print " %i ^ %i = %d" % (base, exponente, potencia)
```

Programa 1.28: Lenguaje Python \Leftarrow Ejercicio 14

Ejercicio 15. Escribir un programa que invierta bit a bit un número ingresado por teclado, ejemplo: si el número ingresado es 5 este se representa de forma binaria en 00101 y el resultado de negar bit a bit este número es 11010 que corresponde a -6

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     int numero;
9
10    cout << "ingrese el valor del numero: ";
11    cin >> numero;
12
13    cout << numero << " = " << ~numero << endl;
14
15    return 0;
16 }
```

Programa 1.29: Lenguaje C \Leftarrow Ejercicio 15

```
1 numero = int(input("ingrese el valor del numero: "))
2
3 print " %i = %i" % (numero, ~numero)
```

Programa 1.30: Lenguaje Python \Leftarrow Ejercicio 15

Ejercicio 16. Escribir un programa que calcule el resultado de duplicar acumulativamente un número leído por teclado la cantidad de veces indicadas por el usuario

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     int numero;
9     int veces;
10
11    cout << "ingrese el valor del numero: ";
12    cin >> numero;
```

```
13
14 cout << "cuantas veces desea duplicarlo: ";
15 cin >> veces;
16
17 cout << numero << " duplicado " << veces << " es " << (numero<<veces
18 ) << endl;
19
20 return 0;
21 }
```

Programa 1.31: Lenguaje C \Leftarrow Ejercicio 16

```
1 numero = int(input("ingrese el valor del numero: "))
2 veces = int(input("cuantas veces desea duplicarlo: "))
3
4 print (" %i duplicado %i veces es %i" % (numero, veces, numero<<
veces))
```

Programa 1.32: Lenguaje Python \Leftarrow Ejercicio 16

Ejercicio 17. Escribir un programa que calcule el valor de sumar dos enteros sin tener en cuenta el arrastre a nivel binario en la suma

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     int numero1;
9     int numero2;
10
11     cout << "ingrese el primer numero: ";
12     cin >> numero1;
13
14     cout << "ingrese el segundo numero: ";
15     cin >> numero2;
16
17     cout << numero1 << " + " << numero2 << " sin arrastre binario es "
18     << (numero1^numero2) << endl;
19
20     return 0;
21 }
```

Programa 1.33: Lenguaje C \Leftarrow Ejercicio 17

```
1 numero1 = int(input("ingrese el primer numero: "))
2 numero2 = int(input("cuantas el segundo numero: "))
3
4 print (" %i + %i sin arrastre binario es %i" % (numero1, numero2,
numero1^numero2))
```

Programa 1.34: Lenguaje Python \Leftarrow Ejercicio 17

Ejercicio 18. Escribir un programa que calcule el residuo de la división entera de dos números leídos por teclado

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     int numero1;
9     int numero2;
10
11     cout << "ingrese el dividendo: ";
12     cin >> numero1;
13
14     cout << "ingrese el divisor: ";
15     cin >> numero2;
16
17     cout << numero1 << " / " << numero2 << " sobran " << (numero1%
18         numero2) << endl;
19
20     return 0;
21 }
```

Programa 1.35: Lenguaje C \Leftarrow Ejercicio 18

```
1 numero1 = int(input("ingrese el dividendo: "))
2 numero2 = int(input("ingrese el divisor: "))
3
4 print (" %i / %i sobran %i" % (numero1, numero2, numero1 % numero
5     2))
```

Programa 1.36: Lenguaje Python \Leftarrow Ejercicio 18

Ejercicio 19. Escribir un programa que lea el radio de un círculo como un número flotante y muestre el área y el perímetro del círculo.

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(int argc, char **argv)
7 {
8     float radio;
9     float area;
10    float perimetro;
11
12    cout << "ingrese el radio: ";
13    cin >> radio;
14
15    perimetro = 2 * 3.1416 * radio;
16    area = 3.1416 * pow(radio, 2);
17
18    cout << " perimetro = " << perimetro << endl;
```

```
19 cout << " area = " << area << endl;
20
21 return 0;
22 }
```

Programa 1.37: Lenguaje C \Leftarrow Ejercicio 19

```
1 radio = float(input("ingrese el radio: "))
2 area = 3.1416 * radio ** 2
3 perimetro = 2 * 3.1416 * radio
4
5 print (" perimetro = %f" % (perimetro))
6 print (" area = %f" % (area))
```

Programa 1.38: Lenguaje Python \Leftarrow Ejercicio 19

Capítulo 2

Estructura de Decisión

2.1. Estructura Si-No

Ejercicio 20. Escribir un programa que detecte si un número introducido desde el teclado es positivo o negativo.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int n;
6     cout << "Ingrese el valor de n: ";
7     cin >> n;
8     if(n >= 0){
9         cout << "el numero leído es positivo " << endl;
10    }else{
11        cout << "el numero leído es negativo " << endl;
12    }
13    return 0;
14 }
```

Programa 2.1: Lenguaje C \Leftarrow Ejercicio 20

```
1 n = int(input("Ingrese el valor de n: "))
2 if n >= 0:
3     print("el numero leído es positivo. ")
4 else:
5     print("el numero leído es negativo. ")
```

Programa 2.2: Lenguaje Python \Leftarrow Ejercicio 20

Ejercicio 21. Escribir un programa que determine si un número leído desde el teclado es par o impar.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int n;
6     cout << "Ingrese el valor de n: ";
7     cin >> n;
8     if(n % 2 == 0){
9         cout << "el numero leído es par " << endl;
10    }else{
11        cout << "el numero leído es impar " << endl;
12    }
13    return 0;
14 }
```

Programa 2.3: Lenguaje C \Leftarrow Ejercicio 21

```
1 n = int(input("Ingrese el valor de n: "))
2 if n % 2 == 0:
3     print("el numero leído es par. ")
4 else:
5     print("el numero leído es impar. ")
```

Programa 2.4: Lenguaje Python \Leftarrow Ejercicio 21

Ejercicio 22. Escribir un programa que una vez leída una hora en formato (horas, minutos, segundos) indique cual será el tiempo dentro de un segundo

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int horas;
8     int minutos;
9     int segundos;
10
11     cout<<"Ingrese la cantidad de horas (Formato 24 horas): ";
12     cin>>horas;
13     cout<<"Ingrese la cantidad de minutos: ";
14     cin>>minutos;
15     cout<<"Ingrese la cantidad de segundos: ";
16     cin>>segundos;
17     if (horas >= 0 && horas <= 23 && minutos >= 0 && minutos <= 59 &&
segundos >= 0 && segundos <= 59) {
18         cout<<"Hora actual: "<<horas<<" : "<<minutos<<" : "<<segundos<<
endl;
19         segundos=segundos+1;
20         if (segundos==60) {
21             segundos=0;
22             minutos=minutos+1;
23             if (minutos==60) {
24                 minutos=0;
```

```
25     horas=horas+1;
26     if (horas==24) {
27         horas=0;
28     }
29 }
30 }
31 cout<<"Hora después de un segundo: "<<horas<<" : "<<minutos<<" : "
<<segundos<<endl;
32 } else {
33     cout<<"La Hora: "<<horas<<" : "<<minutos<<" : "<<segundos<<" es
incorrecta."<<endl;
34 }
35
36 return 0;
37
38 }
```

Programa 2.5: Lenguaje C \Leftarrow Ejercicio 22

```
1 from math import *
2
3 horas = int(input("Ingresa la cantidad de horas (Formato 24 horas): "))
4 minutos = int(input("Ingresa la cantidad de minutos: "))
5 segundos = int(input("Ingresa la cantidad de segundos: "))
6
7 if horas >= 0 and horas <= 23 and minutos >= 0 and minutos <= 59 and
segundos >= 0 and segundos <= 59:
8     print("Hora actual: " + str(horas) + " : " + str(minutos) + " : "
+ str(segundos))
9     segundos = segundos + 1
10    if segundos == 60:
11        segundos = 0;
12        minutos = minutos + 1
13        if minutos == 60:
14            minutos = 0
15            horas = horas + 1
16            if horas == 24:
17                horas = 0
18
19    print("Hora después de un segundo: " + str(horas) + " : " + str(
minutos) + " : " + str(segundos))
20 else:
21    print("La Hora: " + str(horas) + " : " + str(minutos) + " : " +
str(segundos) + " es incorrecta.")
```

Programa 2.6: Lenguaje Python \Leftarrow Ejercicio 22

Ejercicio 23. Escribir un programa que dados dos números enteros positivos y una vez realizadas las comprobaciones necesarias muestre por pantalla uno de los siguientes mensajes:

- El segundo es el cuadrado exacto del primero.
- El segundo es menor que el cuadrado del primero.

- El segundo es mayor que el cuadrado del primero.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int primero;
8     int segundo;
9
10    cout<<"Ingrese el primer número ( >= 0): ";
11    cin>>primero;
12    cout<<"Ingrese el segundo número ( >= 0): ";
13    cin>>segundo;
14    if (primero>=0 && segundo>=0) {
15        if (segundo*segundo==primero)
16            cout<<"El segundo es el cuadrado exacto del primero"<<endl;
17        else if (segundo<primero*primero)
18            cout<<"El segundo es menor que el cuadrado del primero"<<endl;
19        else
20            cout<<"El segundo es mayor que el cuadrado del primero"<<endl;
21    }
22    else
23        cout<<"Error en el ingreso de los datos: "<<primero<<" , "<<
24        segundo<<endl;
25
26    return 0;
27
28 }
```

Programa 2.7: Lenguaje C \Leftarrow Ejercicio 23

```
1 from math import *
2
3 primero = int(input("Ingrese el primer número ( >= 0): "))
4 segundo = int(input("Ingrese el segundo número ( >= 0): "))
5
6 if primero >= 0 and segundo >= 0:
7     if segundo * segundo == primero:
8         print("El segundo es el cuadrado exacto del primero")
9     elif segundo < primero * primero:
10        print("El segundo es menor que el cuadrado del primero")
11    else:
12        print("El segundo es mayor que el cuadrado del primero")
13 else:
14    print("Error en el ingreso de los datos: " + str(primero) + " , "
15        + str(segundo))
```

Programa 2.8: Lenguaje Python \Leftarrow Ejercicio 23

Ejercicio 24. Escribir un programa que calcula el equivalente en grados Fahrenheit o Celsius de una temperatura t , el usuario debe indicar si la temperatura que ingreso esta en celcius o fahrenheit acompañando

el valor por el carácter c o t respectivamente. (Celsius / 5 = (Fahrenheit - 32) * 9)

```
1 #include <iostream>
2 #include <string.h>
3 #include <stdlib.h>
4
5
6 using namespace std;
7
8 int main(){
9     double t1, t2;
10    string entrada;
11    cout << "Ingrese la temperatura que desea convertir. \nEjemplo: (35
    c - 74 f): ";
12    getline(cin, entrada);
13
14
15    if(entrada.find('c') != string::npos){
16        entrada.replace(entrada.find('c'), 1, " ");
17        t1 = atof(entrada.data());
18        t2 = t1 * 9 / 5 + 32;
19        cout << t1 << " C " << " equivale a " << t2 << " F" << endl;
20    }
21    else if(entrada.find('f') != string::npos) {
22        entrada.replace(entrada.find('f'), 1, " ");
23        t1 = atof(entrada.data());
24        t2 = (t1 - 32) * 5 / 9;
25        cout << t1 << " F " << " equivale a " << t2 << " C" << endl;
26    }
27    else cout << "Error en el ingreso de la temperatura: " << entrada
    << endl;
28
29
30    return 0;
31 }
```

Programa 2.9: Lenguaje C \Leftarrow Ejercicio 24

```
1 entrada = input("Ingrese la temperatura que desea convertir. \nEjemplo
: (35 c - 74 f): ")
2
3 if 'c' in entrada:
4     entrada = entrada.replace('c', ' ')
5     t1 = float(entrada)
6     t2 = t1 * 9 / 5 + 32
7     print(str(t1) + " C " + " equivale a " + str(t2) + " F")
8 elif 'f' in entrada:
9     entrada = entrada.replace('f', ' ')
10    t1 = float(entrada)
11    t2 = (t1 - 32) * 5 / 9
12    print(str(t1) + " F " + " equivale a " + str(t2) + " C")
13 else:
14    print("Error en el ingreso de la temperatura: " + str(entrada))
```

Programa 2.10: Lenguaje Python \Leftarrow Ejercicio 24

Ejercicio 25. Escribir un programa que dado un número entre 1 y 12 indique cuantos días tiene el mes correspondiente

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
7     int numero;
8
9     cout << "ingrese el numero a evaluar: ";
10    cin >> numero;
11
12    if(numero == 1 || numero == 3 || numero == 5 || numero == 7 ||
13       numero == 8 || numero == 10 || numero == 12){
14        cout << "31 dias" << endl;
15    }else if(numero == 2){
16        cout << "28 o 29 dias" << endl;
17    }else if(numero == 4 || numero == 6 || numero == 9 || numero == 11){
18        cout << "30 dias" << endl;
19    }else{
20        cout << "numero no valido" << endl;
21    }
22    return 0;
23 }
```

Programa 2.11: Lenguaje C \Leftarrow Ejercicio 25

```
1 numero = int(input("ingrese el numero a evaluar: "))
2
3 if numero == 1 or numero == 3 or numero == 5 or numero == 7 or numero
4   == 8 or numero == 10 or numero == 12:
5     print("31 dias")
6 elif numero == 2:
7     print("28 o 29 dias")
8 elif numero == 4 or numero == 6 or numero == 9 or numero == 11:
9     print("30 dias")
10 else:
11     print("numero no valido")
```

Programa 2.12: Lenguaje Python \Leftarrow Ejercicio 25

Ejercicio 26. Escribir un programa que determine si un año es bisiesto. Un año es bisiesto si es múltiplo de 4. Los años múltiplos de 100 no son bisiestos, salvo si ellos son también múltiplos de 400.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
7     int agno;
8 }
```

```
9  cout << "ingrese el agno a evaluar: ";
10 cin >> agno;
11
12  if(agno % 4 == 0 && (agno % 100 != 0 || agno % 400 == 0)){
13      cout << "bisiesto" << endl;
14  }else{
15      cout << "no bisiesto" << endl;
16  }
17
18  return 0;
19 }
```

Programa 2.13: Lenguaje C \Leftarrow Ejercicio 26

```
1 agno = int(input("ingrese el agno a evaluar: "))
2
3 if agno % 4 == 0 and (agno % 100 != 0 or agno % 400 == 0):
4     print("bisiesto")
5 else:
6     print("no bisiesto")
```

Programa 2.14: Lenguaje Python \Leftarrow Ejercicio 26

Ejercicio 27. Escribir un programa que capture un año, un mes y un día e indique si es una fecha válida o no

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
7     int agno;
8     int mes;
9     int dia;
10    bool bandera;
11
12    cout << "ingrese el agno: ";
13    cin >> agno;
14
15    cout << "ingrese el mes: ";
16    cin >> mes;
17
18    cout << "ingrese el dia: ";
19    cin >> dia;
20
21    bandera = true;
22
23    if(mes < 1 || mes > 12){
24        bandera = false;
25    }else if(mes == 1 || mes == 3 || mes == 5 || mes == 7 || mes == 8 ||
26            mes == 10 || mes == 12){
27        if(dia > 31){
28            bandera = false;
29        }
30    }else if(mes == 2){
```

```
30     if(agno % 4 == 0 && (agno % 100 != 0 || agno % 400 == 0)){
31         if(dia > 29){
32             bandera = false;
33         }
34     }else{
35         if(dia > 28){
36             bandera = false;
37         }
38     }
39 }else if(mes == 4 || mes == 6 || mes == 9 || mes == 11){
40     if(dia > 30){
41         bandera = false;
42     }
43 }
44 if(bandera){
45     cout << agno << "-" << mes << "-" << dia << " es valida" << endl;
46 }else{
47     cout << agno << "-" << mes << "-" << dia << " no es valida" <<
48     endl;
49 }
50 return 0;
51 }
```

Programa 2.15: Lenguaje C \Leftarrow Ejercicio 27

```
1 bandera = True
2
3 agno = int(input("ingrese el agno: "))
4 mes = int(input("ingrese mes: "))
5 dia = int(input("ingrese el dia: "))
6
7 if mes < 1 or mes > 12:
8     bandera = False
9 elif mes == 1 or mes == 3 or mes == 5 or mes == 7 or mes == 8 or mes
10    == 10 or mes == 12:
11     if dia > 31:
12         bandera = False
13 elif mes == 2:
14     if agno % 4 == 0 and (agno % 100 != 0 or agno % 400 == 0):
15         if dia > 29:
16             bandera = False
17     else:
18         if dia > 28:
19             bandera = False
20 elif mes == 4 or mes == 6 or mes == 9 or mes == 11:
21     if dia > 30:
22         bandera = False
23
24 if bandera:
25     print("%i-%i-%i es valida" %(agno,mes,dia))
26 else:
27     print("%i-%i-%i no es valida" %(agno,mes,dia))
```

Programa 2.16: Lenguaje Python \Leftarrow Ejercicio 27

Ejercicio 28. Escribir un programa que le indique al usuario si su peso es adecuado de acuerdo a su estatura, teniendo en cuenta la fórmula de Hamwi: cuando el sexo es femenino es: $45.5 \text{ kg.} + 0.866 * [\text{altura (cm)} - 152.4]$; si el sexo es masculino es: $50 \text{ kg.} + 1.06 * [\text{altura (cm)} - 152.4]$.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
7     int altura;
8     float peso;
9     char sexo;
10
11     cout << "ingrese altura: ";
12     cin >> altura;
13
14     cout << "ingrese su sexo(m/f): ";
15     cin >> sexo;
16
17     if(sexo == 'f'){
18         peso = 45.5 + 0.866 * (altura - 152.4);
19     }else{
20         peso = 50 + 1.06 * (altura - 152.4);
21     }
22
23     cout << "su peso ideal es " << peso << "Kg." << endl;
24     return 0;
25 }
```

Programa 2.17: Lenguaje C \Leftarrow Ejercicio 28

```
1 bandera = True
2
3 altura = int(input("ingrese su altura: "))
4 sexo = input("ingrese su sexo(m/f): ")
5
6 if(sexo == 'f'):
7     peso = 45.5 + 0.866 * (altura - 152.4)
8 else:
9     peso = 50 + 1.06 * (altura - 152.4)
10
11
12 print("su peso ideal es %f Kg." %(peso))
```

Programa 2.18: Lenguaje Python \Leftarrow Ejercicio 28

Ejercicio 29. Escribir un programa que despues de leer 3 números retorne el número medio de los tres. El número medio es aquel que no es el mayor ni el menor de los 3 números

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
7     int altura;
8     float peso;
9     char sexo;
10
11     cout << "ingrese altura: ";
12     cin >> altura;
13
14     cout << "ingrese su sexo(m/f): ";
15     cin >> sexo;
16
17     if(sexo == 'f'){
18         peso = 45.5 + 0.866 * (altura - 152.4);
19     }else{
20         peso = 50 + 1.06 * (altura - 152.4);
21     }
22
23     cout << "su peso ideal es " << peso << "Kg." << endl;
24     return 0;
25 }
```

Programa 2.19: Lenguaje C \Leftarrow Ejercicio 29

```
1 bandera = True
2
3 altura = int(input("ingrese su altura: "))
4 sexo = input("ingrese su sexo(m/f): ")
5
6 if(sexo == 'f'):
7     peso = 45.5 + 0.866 * (altura - 152.4)
8 else:
9     peso = 50 + 1.06 * (altura - 152.4)
10
11
12 print("su peso ideal es %f Kg." %(peso))
```

Programa 2.20: Lenguaje Python \Leftarrow Ejercicio 29

2.2. Estructura de selección

Ejercicio 30. escribir un programa que dado un número del 1 al 12 indique a que mes corresponde y cuantos días tiene el mes.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main(int argc, char **argv)
6 {
```

```
7  int numero;
8
9  cout << "ingrese el numero a evaluar: ";
10 cin >> numero;
11
12 switch(numero){
13     case 1:
14         cout << "Enero tiene 31 dias" << endl;
15         break;
16     case 2:
17         cout << "Febrero tiene 28 o 29 dias" << endl;
18         break;
19     case 3:
20         cout << "Marzo tiene 31 dias" << endl;
21         break;
22     case 4:
23         cout << "Abril tiene 30 dias" << endl;
24         break;
25     case 5:
26         cout << "Mayo tiene 31 dias" << endl;
27         break;
28     case 6:
29         cout << "Junio tiene 300 dias" << endl;
30         break;
31     case 7:
32         cout << "Juio tiene 31 dias" << endl;
33         break;
34     case 8:
35         cout << "Agosto tiene 31 dias" << endl;
36         break;
37     case 9:
38         cout << "Septiembre tiene 30 dias" << endl;
39         break;
40     case 10:
41         cout << "Octubre tiene 31 dias" << endl;
42         break;
43     case 11:
44         cout << "Noviembre tiene 30 dias" << endl;
45         break;
46     case 12:
47         cout << "Diciembre tiene 31 dias" << endl;
48         break;
49     default:
50         cout << "numero no valido" << endl;
51         break;
52 }
53 return 0;
54 }
```

Programa 2.21: Lenguaje C \Leftarrow Ejercicio 30

```
1 numero = int(input("ingrese el numero a evaluar: "))
2
3 if numero == 1:
4     print("Enero tiene 31 dias")
5 elif numero == 2:
6     print("Febrero tiene 28 o 29 dias")
7 elif numero == 3:
8     print("Marzo tiene 31 dias")
```

```
9 elif numero == 4:
10     print("Abril tiene 30 dias")
11 elif numero == 5:
12     print("Mayo tiene 31 dias")
13 elif numero == 6:
14     print("Junio tiene 30 dias")
15 elif numero == 7:
16     print("Julio tiene 31 dias")
17 elif numero == 8:
18     print("Aosto tiene 31 dias")
19 elif numero == 9:
20     print("Septiembre tiene 30 dias")
21 elif numero == 10:
22     print("Outubre tiene 31 dias")
23 elif numero == 11:
24     print("Noviembre tiene 30 dias")
25 elif numero == 12:
26     print("Diciembre tiene 31 dias")
27 else:
28     print("numero no valido")
```

Programa 2.22: Lenguaje Python \Leftarrow Ejercicio 30

Ejercicio 31. Escribir un programa que dado un número del 1 a 7 escriba el correspondiente nombre del día de la semana.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int i;
6     cout << "Ingrese un número ";
7     cin >> i;
8     switch(i){
9     case 1:
10         cout << "el día es lunes" << endl;
11         break;
12     case 2:
13         cout << "el día es martes" << endl;
14         break;
15     case 3:
16         cout << "el día es miércoles" << endl;
17         break;
18     case 4:
19         cout << "el día es jueves" << endl;
20         break;
21     case 5:
22         cout << "el día es viernes" << endl;
23         break;
24     case 6:
25         cout << "el día es sábado" << endl;
26         break;
27     case 7:
28         cout << "el día es domingo" << endl;
29         break;
30     default:
31         cout << "el número no corresponde a un día válido" << endl;
```



```
32     break;
33 }
34
35 return 0;
36 }
```

Programa 2.23: Lenguaje C \Leftarrow Ejercicio 31

```
1 #Python no soporta la estructura switch
2 i = int(input("Ingrese un número  "))
3 if i == 1:
4     print("el día es lunes");
5 elif i == 2:
6     print("el día es martes");
7 elif i == 3:
8     print("el día es miércoles");
9 elif i == 4:
10    print("el día es jueves");
11 elif i == 5:
12    print("el día es viernes");
13 elif i == 6:
14    print("el día es sábado");
15 elif i == 7:
16    print("el día es domingo");
17 else:
18    print("el número no corresponde a un día válido");
```

Programa 2.24: Lenguaje Python \Leftarrow Ejercicio 31

Ejercicio 32. Escribir un programa que lea un carácter e indique si es o no una vocal.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     char c;
6     cout << "Ingrese un carácter: ";
7     cin >> c;
8     switch(c){
9         case 'a':
10        case 'A':
11        case 'e':
12        case 'E':
13        case 'i':
14        case 'I':
15        case 'o':
16        case 'O':
17        case 'u':
18        case 'U':
19            cout << "el carácter es una vocal" << endl;
20            break;
21        default:
22            cout << "el carácter no es una vocal" << endl;
```

```
23     break;
24 }
25
26 return 0;
27 }
```

Programa 2.25: Lenguaje C \Leftarrow Ejercicio 32

```
1 #Python no soporta la estructura switch
2 c = input("Ingrese un carácter: ")
3 if c == 'a' or c == 'A':
4     print("el carácter es una vocal");
5 elif c == 'e' or c == 'E':
6     print("el carácter es una vocal");
7 elif c == 'i' or c == 'I':
8     print("el carácter es una vocal");
9 elif c == 'o' or c == 'O':
10    print("el carácter es una vocal");
11 elif c == 'u' or c == 'U':
12    print("el carácter es una vocal");
13 else:
14    print("el carácter no es una vocal");
```

Programa 2.26: Lenguaje Python \Leftarrow Ejercicio 32

Ejercicio 33. Escribir un programa que permita al usuario elegir el cálculo del área de cualquiera de las figuras geométricas: círculo, triángulo y rectángulo, usando las formulas $\pi * r^2$ (círculo), $(b * h)/2$ (triángulo), $b * h$ (rectángulo). El programa debe presentar un menú al usuario con las siguientes opciones: 1 - círculo 2 - triángulo 3 - rectángulo

```
1 #include <iostream>
2 #include <cmath>
3
4 using namespace std;
5
6 int main(){
7     char c;
8     int base;
9     int altura;
10    float area;
11    int radio;
12
13    cout << "Ingrese una opcion: \n 1 - círculo \n 2 - triángulo \n 3 -
    rectángulo \n";
14    cin >> c;
15
16    switch(c){
17        case '1':
18            cout << "ingrese el valor del radio: ";
19            cin >> radio;
20            area = 3.1416 * pow(radio,2);
21            cout << "el area del círculo es " << area << endl;
```

```
22     break;
23     case '2':
24         cout << "ingrese el valor de la base: ";
25         cin >> base;
26         cout << "ingrese el valor de la altura: ";
27         cin >> altura;
28         area = base * altura / 2;
29         cout << "el area del triangulo es " << area << endl;
30         break;
31     case '3':
32         cout << "ingrese el valor de la base: ";
33         cin >> base;
34         cout << "ingrese el valor de la altura: ";
35         cin >> altura;
36         area = base * altura;
37         cout << "el area del rectangulo es " << area << endl;
38         break;
39     default:
40         cout << "opcion no valida" << endl;
41         break;
42 }
43
44 return 0;
45 }
```

Programa 2.27: Lenguaje C \Leftarrow Ejercicio 33

```
1 #Python no soporta la estructura switch
2
3 opc = input("Ingrese una opcion: \n 1 - circulo \n 2 - triangulo \n 3
- rectangulo \n")
4 if opc == '1':
5     radio = int(input("ingrese el valor del radio: "))
6     area = 3.1416 * (radio ** 2)
7     print("el area del circulo es %f" %(area))
8 elif opc == '2':
9     base = int(input("ingrese el valor de la base: "))
10    altura = int(input("ingrese el valor de la altura: "))
11    area = base * altura / 2
12    print("el area del triangulo es %f" %(area))
13 elif opc == '3':
14    base = int(input("ingrese el valor de la base: "))
15    altura = int(input("ingrese el valor de la altura: "))
16    area = base * altura
17    print("el area del rectangulo es %f" %(area))
18 else:
19    print("opcion no valida");
```

Programa 2.28: Lenguaje Python \Leftarrow Ejercicio 33

Ejercicio 34. Escribir un programa que dado un carácter numérico devuelva su correspondiente número.

```
1 #include <iostream>
2 #include <cmath>
3
```

```
4 using namespace std;
5
6 int main(){
7     char c;
8     int numero;
9
10    cout << "Ingrese un caracter numerico: ";
11    cin >> c;
12
13    switch(c){
14        case '0':
15            numero = 0;
16            break;
17        case '1':
18            numero = 1;
19            break;
20        case '2':
21            numero = 2;
22            break;
23        case '3':
24            numero = 3;
25            break;
26        case '4':
27            numero = 4;
28            break;
29        case '5':
30            numero = 5;
31            break;
32        case '6':
33            numero = 6;
34            break;
35        case '7':
36            numero = 7;
37            break;
38        case '8':
39            numero = 8;
40            break;
41        case '9':
42            numero = 9;
43            break;
44        default:
45            numero = -1;
46            break;
47    }
48
49    if(numero != -1){
50        cout << numero << endl;
51    }else{
52        cout << "caracter no valido" << endl;
53    }
54
55    return 0;
56 }
```

Programa 2.29: Lenguaje C \Leftarrow Ejercicio 34

```
1 #Python no soporta la estructura switch
2
3 opc = input("Ingrese un caracter numerico: ")
```

```
4 if opc == '0':
5     num = 0
6 elif opc == '1':
7     num = 1
8 elif opc == '2':
9     num = 2
10 elif opc == '3':
11     num = 3
12 elif opc == '4':
13     num = 4
14 elif opc == '5':
15     num = 5
16 elif opc == '6':
17     num = 6
18 elif opc == '7':
19     num = 7
20 elif opc == '8':
21     num = 8
22 elif opc == '9':
23     num = 9
24 else:
25     num = -1
26
27 if num != -1:
28     print(num)
29 else:
30     print("caracter no valido")
```

Programa 2.30: Lenguaje Python \Leftarrow Ejercicio 34

Capítulo 3

Estructuras de Iteración

3.1. Iteración

Ejercicio 35. Escribir un programa que visualice en pantalla los números múltiplos de 5 comprendidos entre 1 y 100.

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

Programa 3.1: Lenguaje C \Leftarrow Ejercicio 35

```
1 x = 1
2 while x <= 100:
3     if x % 5 == 0:
4         print(str(x))
5     x = x + 1
```

Programa 3.2: Lenguaje Python \Leftarrow Ejercicio 35

Ejercicio 36. Escribir un programa que capture dos números enteros y muestre en pantalla los números comprendidos entre estos dos números

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int a;
8     int b;
9     int mayor;
10    int menor;
11
12    cout << "Ingrese el primer número: ";
13    cin >> a;
14    cout << "Ingrese el segundo número: ";
15    cin >> b;
16    if (a < b) {
17        menor = a;
18        mayor = b;
19    } else {
20        menor = b;
21        mayor = a;
22    }
23    for (int i = menor+1; i < mayor; i++) {
24        cout << i << " ";
25    }
26
27    return 0;
28 }
```

Programa 3.3: Lenguaje C \Leftarrow Ejercicio 36

```
1 a = int(input("Ingrese el primer numero: "))
2 b = int(input("Ingrese el segundo numero: "))
3
4 if a < b :
5     menor = a
6     mayor = b
7 else :
8     menor = b
9     mayor = a
10
11 for i in range(menor + 1, mayor) :
12     print(str(i),end=' ')
```

Programa 3.4: Lenguaje Python \Leftarrow Ejercicio 36

Ejercicio 37. Escribir un programa que capture dos números enteros y muestre en pantalla los números pares comprendidos entre estos dos números.

```
1 #include<iostream>
2
3 using namespace std;
```

```
4
5 int main() {
6
7     int a;
8     int b;
9     int mayor;
10    int menor;
11
12    cout << "Ingrese el primer número: ";
13    cin >> a;
14    cout << "Ingrese el segundo número: ";
15    cin >> b;
16    if (a < b) {
17        menor = a;
18        mayor = b;
19    } else {
20        menor = b;
21        mayor = a;
22    }
23    for (int i = menor+1; i < mayor; i++) {
24        if (i % 2 == 0)
25            cout << i << " ";
26    }
27
28    return 0;
29 }
```

Programa 3.5: Lenguaje C \Leftarrow Ejercicio 37

```
1 import sys
2 a = int(input("Ingrese el primer número: "))
3 b = int(input("Ingrese el segundo número: "))
4
5 if a < b :
6     menor = a
7     mayor = b
8 else :
9     menor = b
10    mayor = a
11
12 for i in range(menor + 1, mayor) :
13     if i % 2 == 0:
14         sys.stdout.write(str(i) + " ")
```

Programa 3.6: Lenguaje Python \Leftarrow Ejercicio 37

Ejercicio 38. Escribir un programa que genere la tabla de multiplicar de un número introducido por el teclado.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5     int num;
```

```
6  cout<<"Que tabla de multiplicar desea ver? ";
7  cin>>num;
8
9  for (int i=1;i<=10;i++) {
10     cout<<i<<" * "<<num<<" = "<<i*num<<endl;
11 }
12
13 return 0;
14 }
```

Programa 3.7: Lenguaje C \Leftarrow Ejercicio 38

```
1 num = int(input("Que tabla de multiplicar desea ver? "))
2
3 for i in range(1,11):
4     print(str(i) + " * " + str(num) + " = " + str(i*num))
```

Programa 3.8: Lenguaje Python \Leftarrow Ejercicio 38

Ejercicio 39. Escribir un programa que calcule la media de 5 números introducidos por el teclado.

```
1 #include <iostream>
2 using namespace std;
3
4 int main(){
5     int acum=0;
6     int n;
7     for(int x = 0; x < 5; x++){
8         cout << "ingrese el número " << x+1 << " de 5 ";
9         cin >> n;
10        acum += n;
11    }
12    cout << "La media de los 5 números es " << (acum/5.0) << endl;
13    return 0;
14 }
```

Programa 3.9: Lenguaje C \Leftarrow Ejercicio 39

```
1 acum = 0
2 for x in range(0,5):
3     n = int(input("ingrese el número " + str(x+1) + " de 5 " ))
4     acum += n
5 print ("La media de los 5 números es " + str(acum/5.0) )
```

Programa 3.10: Lenguaje Python \Leftarrow Ejercicio 39

Ejercicio 40. Escribir un programa que realice la pregunta desea continuar S/N? y que no deje de hacerla hasta que el usuario teclee N y muestre en pantalla la cantidad de veces que se realizo la pregunta.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     char r;
6     int x=0;
7     do{
8         x++;
9         cout << endl << " desea continuar (S/N) ";
10        cin >> r;
11    }while(r != 'N');
12    cout << "la pregunta se hizo " << x << " veces " << endl;
13    return 0;
14 }
```

Programa 3.11: Lenguaje C \Leftarrow Ejercicio 40

```
1 r = 's'
2 x = 0
3 while (r != 'N'):
4     x += 1
5     r = input(" desea continuar (S/N) ")
6
7 print("la pregunta se hizo " + str(x) + " veces ")
```

Programa 3.12: Lenguaje Python \Leftarrow Ejercicio 40

Ejercicio 41. Escribir un programa que escriba los números comprendidos entre 1 y 1000. El programa escribirá en la pantalla los números en grupos de 20, solicitando al usuario si quiere o no continuar visualizando el siguiente grupo de números.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6     bool continuar;
7     int i;
8     char letra;
9
10    cout<<"Números del 1 al 1000"<<endl;
11    i = 0;
12    continuar = true;
13
14    while (i<=1000 && continuar) {
15        i = i+1;
16        cout << i << " ";
17        if (i % 20 == 0) {
18            cout<<endl;
19            cout<<"Desea continuar? S/s ";
20            cin>>letra;
21            if (!(letra == 's' || letra == 'S')) {
```

```
22     continuar=false;
23     }
24 }
25 }
26 return 0;
27 }
```

Programa 3.13: Lenguaje C \Leftarrow Ejercicio 41

```
1 import sys
2
3 i = 0
4 continuar = True
5
6 while (i <= 1000 and continuar):
7     i = i + 1
8     sys.stdout.write(str(i) + " ")
9     if i % 20 == 0:
10         letra = input("\nDesea continuar? S/s ")
11         if not('s' in letra or 'S' in letra):
12             continuar = False
```

Programa 3.14: Lenguaje Python \Leftarrow Ejercicio 41

Ejercicio 42. Escribir un programa que calcule la media de números introducidos por el teclado hasta que el número ingresado sea cero.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     double media = 0;
8     double num;
9     int total=0;
10
11     do {
12         cout << "Ingrese un número (0 para finalizar): ";
13         cin >> num;
14         media = media + num;
15         total = total + 1;
16     } while (num != 0);
17
18     if( total != 1){
19         media = media/(total-1);
20         cout << "El promedio de los " << (total - 1) << " números
           ingresados es: " << media << endl;
21     }
22     else{
23         cout << "No se ingresaron números." << endl;
24     }
25     return 0;
26 }
```

Programa 3.15: Lenguaje C \Leftarrow Ejercicio 42

```
1 media = 0
2 total = 0
3 num = -1
4
5 while num != 0:
6     num = float(input("Ingrese un número (0 para finalizar): "))
7     media = media + num
8     total = total + 1
9
10 if total != 1:
11     media = media / (total - 1)
12     print("El promedio de los " + str(total - 1) + " números
    ingresados es: " + str(media))
13 else:
14     print("No se ingresaron números. ")
```

Programa 3.16: Lenguaje Python \Leftarrow Ejercicio 42

Ejercicio 43. Escribir un programa que calcule, independientemente, la suma y la media de los números pares e impares comprendidos entre 1 y 200.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int i=1;
8     int sumaimpares = 0;
9     int sumapares = 0;
10
11     cout << "Operaciones sobre los números del 1 al 200." << endl;
12     while (i <= 200) {
13         if (i % 2 == 0) {
14             sumapares=sumapares+i;
15         } else {
16             sumaimpares=sumaimpares+i;
17         }
18         i=i+1;
19     }
20     cout << "Números Pares: " << endl;
21     cout << "La suma es: " << sumapares << endl;
22     cout << "La media es: " << sumapares/100 << endl;
23     cout << "Números ImPares: " << endl;
24     cout << "La suma es: " << sumaimpares << endl;
25     cout << "La media es: " << sumaimpares/100 << endl;
26
27     return 0;
28 }
```

Programa 3.17: Lenguaje C \Leftarrow Ejercicio 43

```
1 print("Operaciones sobre los números del 1 al 200.")
2 i = 1
3 sumaimpares = 0
4 sumapares = 0
5
6 while i <= 200:
7     if i % 2 == 0:
8         sumapares = sumapares + i
9     else:
10        sumaimpares = sumaimpares + i
11    i = i + 1
12
13 print("Números Pares: ")
14 print("La suma es: " + str(sumapares))
15 print("La media es: " + str(sumapares / 100))
16 print("Números ImPares: ")
17 print("La suma es: " + str(sumaimpares))
18 print("La media es: " + str(sumaimpares / 100))
```

Programa 3.18: Lenguaje Python \Leftarrow Ejercicio 43

Ejercicio 44. Escribir un programa que calcule la suma de los cuadrados de los 100 primeros números enteros.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int suma = 0;
8
9     for (int i=1; i <= 100; i++) {
10        suma += i * i;
11    }
12
13    cout << "La suma del cuadrado de los números de 1 a 100 es: " <<
        suma << endl;
14    return 0;
15 }
```

Programa 3.19: Lenguaje C \Leftarrow Ejercicio 44

```
1 suma = 0
2 for i in range(1,101):
3     suma += i * i
4
5 print("La suma del cuadrado de los números de 1 a 100 es: " + str(suma
    ))
```

Programa 3.20: Lenguaje Python \Leftarrow Ejercicio 44

Ejercicio 45. Leer un número entero y determinar si es un número primo.

Número primo: Un número n es primo si es divisible únicamente por la unidad y por el mismo.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6     int i, n;
7     bool tienedivisor=false;
8
9     cout<<"Ingrese un número entero: ";
10    cin>>n;
11
12    if (n>=2) {
13        tienedivisor=false;
14        i=2;
15        while (i<n) {
16            if (n%i==0) {
17                tienedivisor=true;
18                i=n;
19            }
20            i=i+1;
21        }
22        if (!tienedivisor) {
23            cout<<n<<" es un número primo"<<endl;
24        } else {
25            cout<<n<<" no es un número primo"<<endl;
26        }
27    } else {
28        cout<<n<<" debe ser un número mayor o igual a 2"<<endl;
29    }
30    return 0;
31 }
```

Programa 3.21: Lenguaje C \Leftarrow Ejercicio 45

```
1 n = int(input("Ingrese un número entero: "))
2 tienedivisor = False
3
4 if (n >= 2):
5     tienedivisor = False
6
7     if (n >= 2):
8         tienedivisor = False
9         i = 2;
10        while (i < n):
11            if (n % i == 0):
12                tienedivisor = True
13                i = n
14                i = i + 1
15
```

```
16         if (not tienedivisor):
17             print(str(n) + " es un número primo")
18         else:
19             print(str(n) + " no es un número primo")
20     else:
21         print(str(n) + " debe ser un número mayor o igual a 2")
```

Programa 3.22: Lenguaje Python \Leftarrow Ejercicio 45

Ejercicio 46. Imprimir los números perfectos comprendidos en un intervalo dado.

Número perfecto: Un número n es perfecto si la suma de los divisores propios del número es igual a n .

```
1 #include<iostream>
2 #include <time.h>
3
4 using namespace std;
5
6 int main() {
7
8     int a;
9     int b;
10    int mayor;
11    int menor;
12    int sumadivisores;
13    clock_t inicio , fin;
14
15    cout << "Ingrese límite inferior del rango: ";
16    cin >> a;
17    cout << "Ingrese límite superior del rango: ";
18    cin >> b;
19
20    if (a>=1 && b>=1) {
21        if (a<b) {
22            menor = a;
23            mayor = b;
24        } else {
25            menor = b;
26            mayor = a;
27        }
28
29        inicio = clock();
30        for (int i = menor; i <= mayor; i++) {
31            sumadivisores=0;
32            for (int j = 1; j <= i-1; j++) {
33                if (i % j == 0) {
34                    sumadivisores = sumadivisores + j;
35                }
36            }
37            if (sumadivisores == i) {
38                cout << i << " ";
39            }
40        }
41    }
```



```
40     }
41     fin = clock();
42     cout << "\nTiempo de ejecución: " << (fin - inicio) << "
milisegundos.";
43 } else {
44     cout << "Error en el ingreso del intervalo de búsqueda [" << a << ", " <<
b << "]" << endl;
45 }
46
47 return 0;
48 }
```

Programa 3.23: Lenguaje C \Leftarrow Ejercicio 46

```
1 import sys
2 import time
3
4 a = int(input("Ingrese límite inferior del rango: "))
5 b = int(input("Ingrese límite superior del rango: "))
6
7 if a >= 1 and b >= 1:
8     if a < b:
9         menor = a
10        mayor = b
11    else:
12        menor = b
13        mayor = a
14
15    inicio = time.clock();
16    for i in range(menor, mayor+1):
17        sumaDivisores = 0
18        for j in range(1,i):
19            if i % j == 0:
20                sumaDivisores = sumaDivisores + j
21
22        if sumaDivisores == i:
23            sys.stdout.write(str(i) + " ")
24    fin = time.clock();
25    print("\nTiempo de ejecución: " + str(fin - inicio) + "
milisegundos.")
26 else:
27    print("Error en el ingreso del intervalo de búsqueda [" + str(a) +
", " + str(b) + "]" )
```

Programa 3.24: Lenguaje Python \Leftarrow Ejercicio 46

Ejercicio 47. Juego Adivinar número entre 0 y n, con k intentos.

```
1 #include<iostream>
2 #include<cmath>
3 using namespace std;
4
5 int main() {
6
7     bool bandera;
```

```
8  int intento;
9  int limite;
10 int numero;
11 int numintentos;
12
13 cout<<"El número a adivinar esta en el rango [0,limite)." << endl;
14
15 bandera = false;
16 while (!bandera) {
17     cout<<"Ingrese el valor del límite: (>= 1): ";
18     cin>>limite;
19     if (limite >= 1) {
20         bandera=true;
21     }
22 }
23
24 bandera = false;
25 while (!bandera) {
26     cout<<"Ingrese la cantidad de intentos: (>= 1): ";
27     cin>>numintentos;
28     if (numintentos >= 1) {
29         bandera=true;
30     }
31 }
32 srand(time(0));
33 numero = (rand() % limite);
34
35 bandera=false;
36 while (!bandera && numintentos > 0) {
37     cout<<"Tienes "<<numintentos<<" intentos."<<endl;
38     cout<<"Ingrese un número: ";
39     cin>>intento;
40     if (numero == intento) {
41         bandera = true;
42     } else {
43         if (intento < numero) {
44             cout<<intento<<" es menor que el número oculto"<<endl;
45         } else {
46             cout<<intento<<" es mayor que el número oculto"<<endl;
47         }
48     }
49     numintentos = numintentos-1;
50 }
51 if (bandera) {
52     cout<<"Felicidades has ganado."<<endl;
53     cout<<"Aún tenías "<<numintentos<<" intentos"<<endl;
54 } else {
55     cout<<"No lo has logrado!!!!."<<endl;
56     cout<<"El número oculto es: "<<numero<<endl;
57 }
58
59 return 0;
60
61 }
```

Programa 3.25: Lenguaje C \Leftarrow Ejercicio 47

```
1 import random
2
```

```
3 print("El número a adivinar esta en el rango [0,límite).")
4 bandera = False
5 while (not bandera):
6     limite = int(input("Ingrese el valor del límite: (>= 1): "));
7     if limite >= 1:
8         bandera = True
9
10 bandera = False
11 while (not bandera):
12     numintentos = int(input("Ingrese la cantidad de intentos: (>= 1):
13     "));
14     if numintentos >= 1:
15         bandera = True
16
17 numero = random.randint(0, limite)
18
19 bandera = False
20 while not bandera and numintentos > 0:
21     print("Tienes " + str(numintentos)+ " intentos.")
22     intento = int(input("Ingrese un número: "))
23     if numero == intento:
24         bandera = True;
25     elif intento < numero:
26         print(str(intento) + " es menor que el número oculto")
27     else:
28         print(str(intento) + " es mayor que el número oculto")
29     numintentos = numintentos - 1;
30
31 if bandera:
32     print("Felicidades has ganado.")
33     print("Aún tenías " + str(numintentos) + " intentos")
34 else:
35     print("No lo has logrado!!!!.")
36     print("El número oculto es: " + str(numero))
```

Programa 3.26: Lenguaje Python \Leftarrow Ejercicio 47

Ejercicio 48. Escribir un programa que genere la serie de fibonacci hasta el indice que desee el usuario

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int num_a = 0;
8     int num_b = 1;
9     int temp;
10    int indice = 0;
11
12    while(indice <= 2){
13        cout << "hasta que indice desea generar la serie , debe ser mayor a
14        2: " << endl;
15        cin >> indice;
16    }
```

```
17  for(int i=0; i<indice; i++){
18      cout << num_b << " ";
19      temp = num_a;
20      num_a = num_b;
21      num_b = temp + num_a;
22  }
23
24  return 0;
25
26 }
```

Programa 3.27: Lenguaje C \Leftarrow Ejercicio 48

```
1 num_a = 0
2 num_b = 1
3
4 indice = 0
5
6 while indice <= 2:
7     indice = int(input("hasta que indice desea generar la serie, debe
8         ser mayor a 2: "))
9 print(num_a, end=" ")
10 for i in range(0, indice):
11     print(num_b, end=" ")
12     temp = num_a
13     num_a = num_b
14     num_b = temp + num_a
```

Programa 3.28: Lenguaje Python \Leftarrow Ejercicio 48

Ejercicio 49. Escribir un programa que permita multiplicar dos números leídos desde el teclado sin usar el operador de multiplicación

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int num_a;
8     int num_b;
9
10    int acum = 0;
11
12    cout << "ingrese el primer numero: ";
13    cin >> num_a;
14
15    cout << "ingrese el segundo numero: ";
16    cin >> num_b;
17
18    for(int i=0; i < num_b; i++){
19        acum = acum + num_a;
20    }
21
22    cout << num_a << " * " << num_b << " = " << acum << endl;
```

```
23
24     return 0;
25
26 }
```

Programa 3.29: Lenguaje C \Leftarrow Ejercicio 49

```
1 num_a = int(input("ingrese el primer numero: "))
2 num_b = int(input("ingrese el segundo numero: "))
3
4 acum = 0
5
6 for i in range(0, num_b):
7     acum = acum + num_a
8
9 print("%i * %i = %i" % (num_a, num_b, acum))
```

Programa 3.30: Lenguaje Python \Leftarrow Ejercicio 49

Ejercicio 50. Escriba un programa que divida dos numeros leidos sin usar el operador de la división.

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6
7     int num_a;
8     int num_b;
9     int copia_a;
10
11     int cont = 0;
12
13     cout << "ingrese el primer numero: ";
14     cin >> num_a;
15
16     cout << "ingrese el segundo numero: ";
17     cin >> num_b;
18
19     copia_a = num_a;
20
21     while(copia_a - num_b > 0){
22         cont++;
23         copia_a-=num_b;
24     }
25
26     cout << num_a << " / " << num_b << " = " << cont << endl;
27
28     return 0;
29
30 }
```

Programa 3.31: Lenguaje C \Leftarrow Ejercicio 50

```
1 num_a = int(input("ingrese el primer numero: "))
2 num_b = int(input("ingrese el segundo numero: "))
3
4 copia_a = num_a
5
6 cont = 0
7
8 while copia_a - num_b > 0:
9     cont = cont + 1
10    copia_a = copia_a - num_b
11
12 print("%i / %i = %i" %(num_a, num_b, cont))
```

Programa 3.32: Lenguaje Python \Leftarrow Ejercicio 50

Ejercicio 51. Escribir un programa que genere dos columnas de números, la primera con los números del 1 al 10 y la segunda con los números del 10 al 1

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

Programa 3.33: Lenguaje C \Leftarrow Ejercicio 51

```
1 tope = 11
2 for i in range(1,tope):
3     print("%i \t %i" %(i, tope-i))
```

Programa 3.34: Lenguaje Python \Leftarrow Ejercicio 51

Ejercicio 52. Escribir un programa que lea una serie de números desde el teclado hasta que el número ingresado sea cero y muestre en pantalla el mayor número leído, el menor número leído y el promedio de los números leídos (no debe tener en cuenta el cero)

```
1 #include<iostream>
2
3 using namespace std;
4
```

```
5 int main() {
6
7     int mayor = 0;
8     int menor = 0;
9     int contador = 0;
10    int acumulador = 0;
11    int  numero;
12
13    while(true){
14        cout << "ingrese un numero (cero para terminar): ";
15        cin >> numero;
16        if(numero == 0){
17            break;
18        }else{
19            if(numero > mayor || mayor == 0)
20                mayor = numero;
21            if(numero < menor or menor == 0)
22                menor = numero;
23            contador++;
24            acumulador += numero;
25        }
26    }
27    cout << "mayor = " << mayor << endl;
28    cout << "menor = " << menor << endl;
29    cout << "promedio = " << (acumulador/contador) << endl;
30
31    return 0;
32
33 }
```

Programa 3.35: Lenguaje C \Leftarrow Ejercicio 52

```
1 mayor = 0
2 menor = 0
3 contador = 0
4 acumulador = 0
5
6 while True:
7     numero = int(input("ingrese un numero (cero para terminar): "))
8     if numero == 0:
9         break
10    else:
11        if numero > mayor or mayor == 0:
12            mayor = numero
13        if numero < menor or menor == 0:
14            menor = numero
15        contador = contador + 1
16        acumulador = acumulador + numero
17
18 print("mayor = %i" %(mayor))
19 print("menor = %i" %(menor))
20 print("promedio = %i" %(acumulador/contador))
```

Programa 3.36: Lenguaje Python \Leftarrow Ejercicio 52

Ejercicio 53. Escribir un programa que calcule el factorial de un número leído por el teclado

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

Programa 3.37: Lenguaje C \Leftarrow Ejercicio 53

```
1 acumulador = 1
2 numero = int(input("ingrese un numero: "))
3
4 for i in range(1,numero+1):
5     acumulador = acumulador * i
6
7 print("factorial = %i" %(acumulador))
```

Programa 3.38: Lenguaje Python \Leftarrow Ejercicio 53

Ejercicio 54. Escribir un programa que lea un número entero y lo descomponga en sus dígitos, mostrando primero las unidades, luego las decenas, centenas y así hasta terminar los dígitos

```
1 #include<iostream>
2
3 using namespace std;
4
5 int main() {
6     int numero;
7
8     cout << "ingrese un numero: ";
9     cin >> numero;
10
11    while(numero > 0){
12        cout << numero % 10 << endl;
13        numero = int(numero / 10);
14    }
15    return 0;
16 }
17 }
```

Programa 3.39: Lenguaje C \Leftarrow Ejercicio 54

```
1 numero = int(input("ingrese un numero entero: "))
2
3 while numero > 0:
4     print("%i" % (numero % 10))
5     numero = int(numero / 10)
```

Programa 3.40: Lenguaje Python \Leftarrow Ejercicio 54

Capítulo 4

Métodos

4.1. Métodos Iterativos

Ejercicio 55. Escribir un programa que calcule el valor de elevar un número a otro usando un método iterativo.

```
1 #include <iostream>
2 using namespace std;
3
4 long potencia_iterativa(int base, int exponente){
5     long potencia = 1;
6     for(int i = 0; i < exponente; i++){
7         potencia = potencia * base;
8     }
9     return potencia;
10 }
11
12 int main() {
13     int base, exponente;
14     cout << "ingrese la base : ";
15     cin >> base;
16     cout << "ingrese el exponente : ";
17     cin >> exponente;
18     if (exponente >= 0)
19         cout << base << " elevado a " << exponente << " es : " <<
            potencia_iterativa(base, exponente) << endl;
20     else
21         cout << "Error en exponente " << exponente << " debe ser positivo.
            " << endl;
22     return 0;
23 }
```

Programa 4.1: Lenguaje C \Leftarrow Ejercicio 55

```
1 def potencia_iterativa(base, exponente):
2     potencia = 1
3     for i in range(0, exponente):
```

```
4     potencia = potencia * base
5     return potencia
6
7 if __name__ == "__main__":
8     base = int(input("ingrese la base : "))
9     exponente = int(input("ingrese el exponente : "))
10    if (exponente >= 0):
11        print (str(base) + " elevado a " + str(exponente) + " es : " +
12              str(potencia_iterativa(base, exponente)))
13    else:
14        print("Error en exponente " + str(exponente) + " debe ser
15              positivo.")
```

Programa 4.2: Lenguaje Python \Leftarrow Ejercicio 55

Ejercicio 56. Escribir un programa que calcule el factorial de un número usando un método iterativo.

```
1 #include <iostream>
2 using namespace std;
3
4 long factorial_iterativo(int numero){
5     long factorial = 1;
6     for(int i = 1; i <= numero; i++){
7         factorial *= i;
8     }
9     return factorial;
10 }
11
12 int main() {
13     int numero;
14     cout << "ingrese el numero : ";
15     cin >> numero;
16
17     if (numero >= 0)
18         cout << "el factorial de " << numero << " es : " <<
19         factorial_iterativo(numero) << endl;
20     else
21         cout << "el factorial solo se puede calcular a numeros enteros
22         positivos " << endl;
23
24     return 0;
25 }
```

Programa 4.3: Lenguaje C \Leftarrow Ejercicio 56

```
1 def factorial_iterativo(numero):
2     factorial = 1
3     for i in range(1,numero):
4         factorial *= i
5     return factorial
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
```

```
9  if (numero >= 0):
10     print ("el factorial de " + str(numero) + " es : " + str(
        factorial_iterativo(numero)))
11  else:
12     print("el factorial solo se puede calcular a numeros enteros
        positivos ")
```

Programa 4.4: Lenguaje Python \Leftarrow Ejercicio 56

Ejercicio 57. Escribir un programa que calcule el término n de la serie de fibonacci usando un método iterativo.

```
1 #include <iostream>
2 using namespace std;
3
4 long fibonacci_iterativo(int numero){
5     int fibo1 = 0;
6     int fibo2 = 1;
7     int temp;
8     if(numero == 0)
9         return fibo1;
10    else if(numero == 1)
11        return fibo2;
12    else{
13        for(int i = 1; i < numero; i++){
14            temp = fibo2;
15            fibo2 = fibo1 + fibo2;
16            fibo1 = temp;
17        }
18        return fibo2;
19    }
20 }
21
22 int main() {
23     int numero;
24     cout << "ingrese el termino de la serie que desea obtener : ";
25     cin >> numero;
26
27     if (numero >= 0)
28         cout << "el termino " << numero << " de la serie de fibonacci es :
                " << fibonacci_iterativo(numero) << endl;
29     else
30         cout << "la serie de fibonacci solo se genera a partir del
                termino cero" << endl;
31
32     return 0;
33 }
```

Programa 4.5: Lenguaje C \Leftarrow Ejercicio 57

```
1 def fibonacci_iterativo(numero):
2     fibo1 = 0
3     fibo2 = 1
4     temp = 0
```

```
5  if numero == 0:
6      return fibo1
7  elif numero == 1:
8      return fibo2
9  else:
10     for i in range(1,numero):
11         temp = fibo2
12         fibo2 = fibo1 + fibo2
13         fibo1 = temp
14     return fibo2
15
16 if __name__ == "__main__":
17     numero = int(input("ingrese el termino de la serie que desea
18 obtener : "))
19     if (numero >= 0):
20         print ("el termino " + str(numero) + " de la serie de fibonacci
21 es : " + str(fibonacci_iterativo(numero)))
22     else:
23         print("la serie de fibonacci solo se genera a partir del
24 termino cero ")
```

Programa 4.6: Lenguaje Python \Leftarrow Ejercicio 57

Ejercicio 58. Escribir un programa que usando métodos iterativos cuente cuantos dígitos tiene un número entero.

```
1 #include <iostream>
2 using namespace std;
3
4 long cuenta_digitos_iterativo(int numero){
5     int contador = 0;
6     if(numero < 0)
7         numero *= -1;
8     if(numero == 0)
9         contador = 1;
10    while(numero > 0){
11        contador++;
12        numero /= 10;
13    }
14    return contador;
15 }
16
17 int main() {
18     int numero;
19     cout << "ingrese el numero : ";
20     cin >> numero;
21
22     cout << numero << " tiene : " << cuenta_digitos_iterativo(numero) <<
23         " digitos " << endl;
24
25     return 0;
26 }
```

Programa 4.7: Lenguaje C \Leftarrow Ejercicio 58

```
1 def cuenta_digitos_iterativo(numero):
2     contador = 0
3     if numero < 0:
4         numero *= -1
5     if numero == 0:
6         contador = 1
7     while numero > 0:
8         contador += 1
9         numero = int(numero / 10)
10    return contador
11
12 if __name__ == "__main__":
13     numero = int(input("ingrese el numero : "))
14     print (str(numero) + " tiene : " + str(cuenta_digitos_iterativo(
        numero)) + " digitos " )
```

Programa 4.8: Lenguaje Python \Leftarrow Ejercicio 58

Ejercicio 59. Escribir un programa que lea un número entero y lea la base en el cual esta expresado e indique si esto es correcto, el programa debe usar métodos iterativos.

```
1 #include <iostream>
2 using namespace std;
3
4 bool validador_iterativo(int numero, int base){
5     int digito;
6     while(numero > 0){
7         digito = numero % 10;
8         if(digito >= base)
9             return false;
10        numero /= 10;
11    }
12    return true;
13 }
14
15 int main() {
16     int numero, base;
17     cout << "ingrese el numero : ";
18     cin >> numero;
19     cout << "ingrese la base : ";
20     cin >> base;
21
22     if(validador_iterativo(numero, base))
23         cout << "el " << numero << " es valido en base : " << base << endl
24         ;
25     else
26         cout << "el " << numero << " no es valido en base : " << base <<
27         endl;
28     return 0;
29 }
```

Programa 4.9: Lenguaje C \Leftarrow Ejercicio 59

```
1 def validador_iterativo(numero, base):
2     while numero > 0:
3         digito = numero % 10
4         numero = int(numero / 10)
5         if digito >= base:
6             return False
7     return True
8
9 if __name__ == "__main__":
10     numero = int(input("ingrese el numero : "))
11     base = int(input("ingrese la base : "))
12     if validador_iterativo(numero, base):
13         print (str(numero) + " es valido en base " + str(base))
14     else:
15         print (str(numero) + " no es valido en base " + str(base))
```

Programa 4.10: Lenguaje Python \Leftarrow Ejercicio 59

Ejercicio 60. Escribir un programa que usando métodos iterativos halle la suma de los primeros "n" números naturales.

```
1 #include <iostream>
2 using namespace std;
3
4 long sumador_iterativo(int numero){
5     long suma = 0;
6     for(int i = 0; i <= numero; i++){
7         suma += i;
8     }
9     return suma;
10 }
11
12 int main() {
13     int numero;
14     cout << "ingrese el numero : ";
15     cin >> numero;
16
17     cout << "la suma de los " << numero << " primeros numeros naturales
18     es : " << sumador_iterativo(numero) << endl;
19
20     return 0;
21 }
```

Programa 4.11: Lenguaje C \Leftarrow Ejercicio 60

```
1 def sumador_iterativo(numero):
2     suma = 0
3     for i in range(0,numero+1):
4         suma += i
5     return suma
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
```

```
9 print ("la suma de los primeros " + str(numero) + " numeros enteros
    es : " + str(sumador_iterativo(numero)))
```

Programa 4.12: Lenguaje Python \Leftarrow Ejercicio 60

Ejercicio 61. Escribir un programa que usando métodos iterativos verifique si un número leído desde el teclado es o no un número palíndromo.

```
1 #include <iostream>
2 using namespace std;
3
4 int inversor_iterativo(int numero){
5     int temp = 0;
6     while(numero > 0){
7         temp = (temp * 10) + (numero % 10);
8         numero /= 10;
9     }
10    return temp;
11 }
12
13 int main() {
14     int numero;
15     cout << "ingrese el numero : ";
16     cin >> numero;
17
18     if(numero == inversor_iterativo(numero))
19         cout << "el " << numero << " es un palindromo " << endl;
20     else
21         cout << "el " << numero << " no es un palindromo " << endl;
22
23     return 0;
24 }
```

Programa 4.13: Lenguaje C \Leftarrow Ejercicio 61

```
1 def inversor_iterativo(numero):
2     temp = 0
3     while numero > 0:
4         temp = (temp * 10) + (numero % 10)
5         numero = int(numero/10)
6     return temp
7
8 if __name__ == "__main__":
9     numero = int(input("ingrese el numero : "))
10    if(numero == inversor_iterativo(numero)):
11        print ("el " + str(numero) + " es un numero palindromo")
12    else:
13        print ("el " + str(numero) + " no es un numero palindromo")
```

Programa 4.14: Lenguaje Python \Leftarrow Ejercicio 61

Ejercicio 62. Escribir un programa que use un método que permita validar una fecha en formato "YYYY – MM – DD", tenga en cuenta la existencia de años biciestos, es decir que solo se pueden validar fechas desde el año 1582.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <cstring>
4 using namespace std;
5
6 bool biciesto(int anio){
7     if((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0))
8         return true;
9     else
10        return false;
11
12 }
13
14 int dias_mes(int mes, int anio){
15     switch(mes){
16         case 1:
17             return 31;
18         case 2:
19             if(biciesto(anio))
20                 return 29;
21             else
22                 return 28;
23         case 3:
24             return 31;
25         case 4:
26             return 30;
27         case 5:
28             return 31;
29         case 6:
30             return 30;
31         case 7:
32             return 31;
33         case 8:
34             return 31;
35         case 9:
36             return 30;
37         case 10:
38             return 31;
39         case 11:
40             return 30;
41         case 12:
42             return 31;
43     }
44 }
45 bool validar_fecha(string fecha){
46     int anio = atoi(fecha.substr(0,4).c_str());
47     int mes = atoi(fecha.substr(5,2).c_str());
48     int dia = atoi(fecha.substr(8,2).c_str());
49
50     if(anio < 1582)
51         return false;
52     if(mes < 1 || mes > 12)
53         return false;
```

```
54  if(dia < 1 || dia > dias_mes(mes, anio))
55      return false;
56
57  return true;
58
59 }
60
61 int main() {
62     string fecha;
63     cout << "ingrese la fecha : " << endl;
64     cin >> fecha;
65     if(validar_fecha(fecha))
66         cout << fecha << " es valida " << endl;
67     else
68         cout << fecha << " no es valida " << endl;
69
70     return 0;
71 }
```

Programa 4.15: Lenguaje C \Leftarrow Ejercicio 62

```
1 def biciesto(anio):
2     if(anio % 4 == 0 and anio % 100 != 0) or (anio % 400 == 0):
3         return True
4     else:
5         return False
6
7 def dias_mes(mes, anio):
8     if mes == 1:
9         return 31
10    if mes == 2:
11        if biciesto(anio):
12            return 29
13        else:
14            return 28
15    if mes == 3:
16        return 31
17    if mes == 4:
18        return 30
19    if mes == 5:
20        return 31
21    if mes == 6:
22        return 30
23    if mes == 7:
24        return 31
25    if mes == 8:
26        return 31
27    if mes == 9:
28        return 30
29    if mes == 10:
30        return 31
31    if mes == 11:
32        return 30
33    if mes == 12:
34        return 31
35
36 def validar_fecha(fecha):
37     anio = int(fecha[0:4])
38     mes = int(fecha[5:7])
```

```
39     dia = int( fecha[8:10])
40
41     if( anio < 1582):
42         return False
43     if( mes < 1 or mes > 12):
44         return False
45     if( dia < 1 or dia > dias_mes( mes, anio)):
46         return False
47
48     return True
49
50 if __name__ == "__main__":
51     fecha = input("ingrese la fecha ")
52     if validar_fecha( fecha):
53         print( fecha + " es valida")
54     else:
55         print( fecha + " no es valida")
```

Programa 4.16: Lenguaje Python \Leftarrow Ejercicio 62

Ejercicio 63. Escribir un programa que usando métodos permita indicar entre dos fechas capturadas en formato "YYYY-MM-DD" cual es previa.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <cstring>
4 using namespace std;
5
6 bool biciesto(int anio){
7     if((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0))
8         return true;
9     else
10         return false;
11 }
12
13
14 int dias_mes(int mes, int anio){
15     switch(mes){
16         case 1:
17             return 31;
18         case 2:
19             if(biciesto(anio))
20                 return 29;
21             else
22                 return 28;
23         case 3:
24             return 31;
25         case 4:
26             return 30;
27         case 5:
28             return 31;
29         case 6:
30             return 30;
31         case 7:
32             return 31;
```

```
33     case 8:
34         return 31;
35     case 9:
36         return 30;
37     case 10:
38         return 31;
39     case 11:
40         return 30;
41     case 12:
42         return 31;
43 }
44 }
45 bool validar_fecha(string fecha){
46     int anio = atoi(fecha.substr(0,4).c_str());
47     int mes = atoi(fecha.substr(5,2).c_str());
48     int dia = atoi(fecha.substr(8,2).c_str());
49
50     if(anio < 1582)
51         return false;
52     if(mes < 1 || mes > 12)
53         return false;
54     if(dia < 1 || dia > dias_mes(mes, anio))
55         return false;
56
57     return true;
58 }
59 }
60
61 string fecha_anterior(string fecha1, string fecha2){
62     int anio1 = atoi(fecha1.substr(0,4).c_str());
63     int mes1 = atoi(fecha1.substr(5,2).c_str());
64     int dia1 = atoi(fecha1.substr(8,2).c_str());
65     int anio2 = atoi(fecha2.substr(0,4).c_str());
66     int mes2 = atoi(fecha2.substr(5,2).c_str());
67     int dia2 = atoi(fecha2.substr(8,2).c_str());
68
69     if(anio1 < anio2)
70         return fecha1;
71     else if(anio1 == anio2){
72         if(mes1 < mes2)
73             return fecha1;
74         else if(mes1 == mes2){
75             if(dia1 < dia2)
76                 return fecha1;
77             else if(dia1 == dia2)
78                 return "iguales";
79             else
80                 return fecha2;
81         }else
82             return fecha2;
83     }else
84         return fecha2;
85 }
86
87 int main() {
88     string fecha1, fecha2;
89     cout << "ingrese la primera fecha : " << endl;
90     cin >> fecha1;
91     cout << "ingrese la segunda fecha : " << endl;
```

```
92  cin >> fecha2;
93  if(validar_fecha(fecha1) && validar_fecha(fecha2))
94      if(fecha_anterior(fecha1, fecha2) == "iguales")
95          cout << fecha1 << " es igual a " << fecha2 << endl;
96      else if(fecha_anterior(fecha1, fecha2) == fecha1)
97          cout << fecha1 << " es previa a " << fecha2 << endl;
98      else
99          cout << fecha2 << " es previa a " << fecha1 << endl;
100 else
101     cout << " fechas no validas " << endl;
102
103 return 0;
104 }
```

Programa 4.17: Lenguaje C \Leftarrow Ejercicio 63

```
1 def biciesto(anio):
2     if(anio % 4 == 0 and anio % 100 != 0) or (anio % 400 == 0):
3         return True
4     else:
5         return False
6
7 def dias_mes(mes, anio):
8     if mes == 1:
9         return 31
10    if mes == 2:
11        if biciesto(anio):
12            return 29
13        else:
14            return 28
15    if mes == 3:
16        return 31
17    if mes == 4:
18        return 30
19    if mes == 5:
20        return 31
21    if mes == 6:
22        return 30
23    if mes == 7:
24        return 31
25    if mes == 8:
26        return 31
27    if mes == 9:
28        return 30
29    if mes == 10:
30        return 31
31    if mes == 11:
32        return 30
33    if mes == 12:
34        return 31
35
36 def validar_fecha(fecha):
37     anio = int(fecha[0:4])
38     mes = int(fecha[5:7])
39     dia = int(fecha[8:10])
40
41     if(anio < 1582):
42         return False
43     if(mes < 1 or mes > 12):
```

```
44         return False
45     if (dia < 1 or dia > dias_mes(mes, anio)):
46         return False
47     return True
48
49 def fecha_anterior(fecha1, fecha2):
50     anio1 = int(fecha1[0:4])
51     mes1 = int(fecha1[5:7])
52     dia1 = int(fecha1[8:10])
53     anio2 = int(fecha2[0:4])
54     mes2 = int(fecha2[5:7])
55     dia2 = int(fecha2[8:10])
56
57     if (anio1 < anio2):
58         return fecha1
59     elif (anio1 == anio2):
60         if (mes1 < mes2):
61             return fecha1
62         elif (mes1 == mes2):
63             if (dia1 < dia2):
64                 return fecha1
65             elif (dia1 == dia2):
66                 return "iguales"
67             else:
68                 return fecha2
69         else:
70             return fecha2
71     else:
72         return fecha2
73
74 if __name__ == "__main__":
75     fecha1 = input("ingrese la primera fecha ")
76     fecha2 = input("ingrese la segunda fecha ")
77     if (validar_fecha(fecha1) and validar_fecha(fecha2)):
78         if (fecha_anterior(fecha1, fecha2) == "iguales"):
79             print(fecha1 + " es igual a " + fecha2)
80         elif (fecha_anterior(fecha1, fecha2) == fecha1):
81             print(fecha1 + " es previa a " + fecha2)
82         else:
83             print(fecha2 + " es previa a " + fecha1)
84     else:
85         print("fechas no es validas")
```

Programa 4.18: Lenguaje Python \Leftarrow Ejercicio 63

4.2. Métodos Recursivos

Ejercicio 64. Escribir un programa que usando recursividad calcule la suma de los dígitos de un número entero positivo leído desde el teclado.

```
1 #include <iostream>
2 using namespace std;
3
4 long suma_digitos_recursivo(int numero){
```

```
5  if(numero == 0)
6      return 0;
7  else
8      return numero % 10 + suma_digitos_recursivo(numero/10);
9  }
10
11
12 int main() {
13     int numero;
14     cout << "ingrese un numero entero : ";
15     cin >> numero;
16     if (numero >= 0)
17         cout << "los digitos de " << numero << " suman " <<
18             suma_digitos_recursivo(numero) << endl;
19     else
20         cout << "El numero debe ser positivo." << endl;
21     return 0;
22 }
```

Programa 4.19: Lenguaje C \Leftarrow Ejercicio 64

```
1 def suma_digitos_recursivo(numero):
2     if (numero == 0):
3         return 0
4     else:
5         return (numero % 10 + suma_digitos_recursivo(int(numero/10)))
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
9     if (numero >= 0):
10         print ("la suma de los digitos de " + str(numero) + " suman " +
11               str(suma_digitos_recursivo(numero)))
12     else:
13         print("El numero debe ser positivo.")
```

Programa 4.20: Lenguaje Python \Leftarrow Ejercicio 64

Ejercicio 65. Escribir un programa que calcule el valor de elevar un número a otro usando un método recursivo.

```
1 #include <iostream>
2 using namespace std;
3
4 long potencia_recursiva(int base, int exponente){
5     if(exponente == 0)
6         return 1;
7     else
8         return base * potencia_recursiva(base, exponente-1);
9 }
10
11
12 int main() {
13     int base, exponente;
14     cout << "ingrese la base : ";
```



```
15  cin >> base;
16  cout << "ingrese el exponente : ";
17  cin >> exponente;
18  if (exponente >= 0)
19      cout << base << " elevado a " << exponente << " es : " <<
        potencia_recursiva(base, exponente) << endl;
20  else
21      cout << "Error en exponente " << exponente << " debe ser positivo.
        " << endl;
22  return 0;
23 }
```

Programa 4.21: Lenguaje C \Leftarrow Ejercicio 65

```
1 def potencia_recursiva(base, exponente):
2     if (exponente == 0):
3         return 1
4     else:
5         return (base * potencia_recursiva(base, exponente-1))
6
7 if __name__ == "__main__":
8     base = int(input("ingrese la base : "))
9     exponente = int(input("ingrese el exponente : "))
10    if (exponente >= 0):
11        print (str(base) + " elevado a " + str(exponente) + " es : " +
            str(potencia_recursiva(base, exponente)))
12    else:
13        print("Error en exponente " + str(exponente) + " debe ser
            positivo.")
```

Programa 4.22: Lenguaje Python \Leftarrow Ejercicio 65

Ejercicio 66. Escribir un programa que permita calcular el n-esimo termino de la serie de fibonacci de manera recursiva.

```
1 #include <iostream>
2 using namespace std;
3
4 int fibo_recursivo(int termino){
5     if(termino < 2)
6         return termino;
7     else
8         return fibo_recursivo(termino-1) + fibo_recursivo(termino-2);
9 }
10
11
12 int main() {
13     int numero;
14     cout << "ingrese un numero entero : ";
15     cin >> numero;
16     if (numero >= 0)
17         cout << "el termino " << numero << " de la serie de fibonacci es "
            << fibo_recursivo(numero) << endl;
18     else
19         cout << "El numero debe ser positivo." << endl;
20     return 0;
}
```

21 }

Programa 4.23: Lenguaje C \Leftarrow Ejercicio 66

```
1 def fibo_recursivo(termino):
2     if (termino < 2):
3         return termino
4     else:
5         return fibo_recursivo(termino - 1) + fibo_recursivo(termino - 2)
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
9     if (numero >= 0):
10        print ("el termino " + str(numero) + " de la serie de fibonacci
11              es " + str(fibo_recursivo(numero)))
12    else:
13        print("El numero debe ser positivo.")
```

Programa 4.24: Lenguaje Python \Leftarrow Ejercicio 66

Ejercicio 67. Escribir un programa que usando métodos recursivos permita calcular el producto de un numero por otro, no se puede usar el operador de multiplicación para este ejercicio.

```
1 #include <iostream>
2 using namespace std;
3
4 int producto_recursivo(int numero, int multiplicador){
5     if(multiplicador == 0)
6         return 0;
7     else if(multiplicador == 1)
8         return numero;
9     else
10        return numero + producto_recursivo(numero, multiplicador - 1);
11 }
12
13
14 int main() {
15     int numero, multiplicador;
16     cout << "ingrese un numero entero : ";
17     cin >> numero;
18     cout << "ingrese un numero entero : ";
19     cin >> multiplicador;
20     if (multiplicador >= 0)
21         cout << numero << " * " << multiplicador << " = " <<
22         producto_recursivo(numero, multiplicador) << endl;
23     else
24         cout << "El segundo numero debe ser positivo." << endl;
25     return 0;
26 }
```

Programa 4.25: Lenguaje C \Leftarrow Ejercicio 67

```
1 def producto_recursivo(numero, multiplicador):
2     if (multiplicador == 0):
```

```
3     return 0
4     if (multiplicador == 1):
5         return numero
6     else:
7         return numero + producto_recursivo(numero, multiplicador - 1)
8
9 if __name__ == "__main__":
10     numero = int(input("ingrese el numero : "))
11     multiplicador = int(input("ingrese otro numero : "))
12     if (multiplicador >= 0):
13         print (str(numero) + " * " + str(multiplicador) + " = " + str(
14             producto_recursivo(numero, multiplicador)))
15     else:
16         print("El segundo numero debe ser positivo.")
```

Programa 4.26: Lenguaje Python \Leftarrow Ejercicio 67

Ejercicio 68. Escribir un programa que usando métodos recursivos permita calcular el cociente de un numero por otro, no se puede usar el operador de división para este ejercicio.

```
1 #include <iostream>
2 using namespace std;
3
4 int divisor_recursivo(int numero, int divisor){
5     if(divisor > numero)
6         return 0;
7     else
8         return 1 + divisor_recursivo(numero - divisor , divisor);
9 }
10
11
12 int main() {
13     int numero, divisor;
14     cout << "ingrese un numero entero : ";
15     cin >> numero;
16     cout << "ingrese el divisor : ";
17     cin >> divisor;
18     if (divisor > 0)
19         cout << numero << " / " << divisor << " = " << divisor_recursivo(
20             numero, divisor) << endl;
21     else
22         cout << "El divisor debe ser positivo." << endl;
23     return 0;
24 }
```

Programa 4.27: Lenguaje C \Leftarrow Ejercicio 68

```
1 def divisor_recursivo(numero, divisor):
2     if (numero < divisor):
3         return 0
4     else:
5         return 1 + divisor_recursivo(numero - divisor , divisor)
6
7 if __name__ == "__main__":
```

```
8  numero = int(input("ingrese el numero : "))
9  divisor = int(input("ingrese el divisor : "))
10 if (divisor > 0):
11     print (str(numero) + " / " + str(divisor) + " = " + str(
12         divisor_recursivo(numero, divisor))
13 else:
14     print("El divisor debe ser positivo.")
```

Programa 4.28: Lenguaje Python \Leftarrow Ejercicio 68

Ejercicio 69. Escribir un programa que usando métodos recursivos permita calcular el factorial de un número.

```
1 #include <iostream>
2 using namespace std;
3
4 int factorial_recursivo(int numero){
5     if(numero < 1)
6         return 1;
7     else
8         return numero * factorial_recursivo(numero - 1);
9 }
10
11
12 int main() {
13     int numero;
14     cout << "ingrese un numero entero : ";
15     cin >> numero;
16
17     if (numero > 0)
18         cout << numero << "! " << " = " << factorial_recursivo(numero) <<
19         endl;
20     else
21         cout << "El numero debe ser positivo." << endl;
22     return 0;
23 }
```

Programa 4.29: Lenguaje C \Leftarrow Ejercicio 69

```
1 def factorial_recursivo(numero):
2     if (numero < 1):
3         return 1
4     else:
5         return numero * factorial_recursivo(numero - 1)
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
9     if (numero > 0):
10         print (str(numero) + "!" + " = " + str(factorial_recursivo(
11             numero)))
12     else:
13         print("El numero debe ser positivo.")
```

Programa 4.30: Lenguaje Python \Leftarrow Ejercicio 69

Ejercicio 70. Escribir un programa que calcule el máximo común divisor (MCD) de dos enteros positivos. Si $M \geq N$ una función recursiva para MCD es:

$$\begin{aligned} \text{MCD} &= M \text{ si } N = 0 \\ \text{MCD} &= \text{MCD}(N, M \bmod N) \text{ si } N < > 0 \end{aligned}$$

En caso de que los valores ingresados cumplan que $M < N$ el programa debe intercambiar los valores.

```
1 #include <iostream>
2 using namespace std;
3
4 int mcd_recursivo(int m, int n){
5     if(n == 0)
6         return m;
7     else
8         return mcd_recursivo(n, m % n);
9 }
10
11
12 int main() {
13     int numero1, numero2;
14     cout << "ingrese un numero entero : ";
15     cin >> numero1;
16     cout << "ingrese otro numero entero : ";
17     cin >> numero2;
18
19     if (numero2 > numero1){
20         int temp = numero2;
21         numero2 = numero1;
22         numero1 = temp;
23     }
24     cout << "MCD(" << numero1 << ", " << numero2 << ") = " <<
        mcd_recursivo(numero1, numero2) << endl;
25     return 0;
26 }
```

Programa 4.31: Lenguaje C \Leftarrow Ejercicio 70

```
1 def mcd_recursivo(m, n):
2     if (n == 0):
3         return m
4     else:
5         return mcd_recursivo(n, m % n)
6
7 if __name__ == "__main__":
8     numero1 = int(input("ingrese el numero : "))
9     numero2 = int(input("ingrese otro numero : "))
10    if (numero2 > numero1):
11        temp = numero2
12        numero2 = numero1
13        numero1 = temp
14    else:
```

```
15 print("MCD(" + str(numero1) + ", " + str(numero2) + ") = " + str  
    (mcd_recurso(numero1, numero2)))
```

Programa 4.32: Lenguaje Python \Leftarrow Ejercicio 70

Ejercicio 71. Escribir un programa que usando un método recursivo transforme un número entero positivo a notación binaria, debe usar únicamente tipos de datos enteros.

```
1 #include <iostream>  
2 using namespace std;  
3  
4 int a_binario_recursivo(int numero){  
5     if(numero == 0)  
6         return 0;  
7     else  
8         return (numero % 2) + (a_binario_recursivo(numero/2) * 10);  
9 }  
10  
11  
12 int main() {  
13     int numero;  
14     cout << "ingrese un numero entero : ";  
15     cin >> numero;  
16  
17     if (numero >= 0)  
18         cout << a_binario_recursivo(numero);  
19     else  
20         cout << "Error el numero debe ser un entero positivo." << endl;  
21     return 0;  
22 }
```

Programa 4.33: Lenguaje C \Leftarrow Ejercicio 71

```
1 def a_binario_recursivo(numero):  
2     if (numero == 0):  
3         return 0  
4     else:  
5         return (numero % 2) + (a_binario_recursivo(int(numero / 2)) * 1  
6         0)  
7  
8 if __name__ == "__main__":  
9     numero = int(input("ingrese el numero : "))  
10  
11     if (numero > 0):  
12         print(str(numero) + " en binario es : " + str(  
13             a_binario_recursivo(numero)))  
14     else:  
15         print("Error el numero debe ser un entero positivo.")
```

Programa 4.34: Lenguaje Python \Leftarrow Ejercicio 71

Ejercicio 72. Escribir un programa que usando un método recursivo transforme un número en notación binaria a un entero positivo en base 10, debe usar únicamente tipos de datos enteros.

```
1 #include <iostream>
2 using namespace std;
3
4 int binario_a_decimal_recursivo(int numero){
5     if(numero == 0)
6         return 0;
7     if(numero == 1)
8         return 1;
9     else
10        return (numero % 10) + (binario_a_decimal_recursivo(numero/10) *
11        2);
12 }
13
14 int main() {
15     int numero;
16     cout << "ingrese un numero entero : ";
17     cin >> numero;
18
19     if (numero >= 0)
20         cout << binario_a_decimal_recursivo(numero);
21     else
22         cout << "Error el numero debe ser un entero positivo." << endl;
23     return 0;
24 }
```

Programa 4.35: Lenguaje C \Leftarrow Ejercicio 72

```
1 def binario_a_decimal_recursivo(numero):
2     if (numero == 0):
3         return 0
4     if (numero == 1):
5         return 1
6     else:
7         return (numero % 10) + (binario_a_decimal_recursivo(int(numero
8         / 10)) * 2)
9
10 if __name__ == "__main__":
11     numero = int(input("ingrese el numero : "))
12
13     if (numero > 0):
14         print(str(numero) + " en decimal es : " + str(
15             binario_a_decimal_recursivo(numero)))
16     else:
17         print("Error el numero debe ser un entero positivo.")
```

Programa 4.36: Lenguaje Python \Leftarrow Ejercicio 72

Ejercicio 73. Escribir un programa que usando métodos iterativos indique al usuario cuantos dígitos tiene un número entero positivo leído desde el teclado.

```
1 #include <iostream>
2 using namespace std;
3
4 int contador_recursivo(int numero){
5     if(numero == 0)
6         return 0;
7     else
8         return 1 + contador_recursivo(numero/10);
9 }
10
11
12 int main() {
13     int numero;
14     cout << "ingrese un numero entero : ";
15     cin >> numero;
16     if (numero >= 0)
17         cout << "el numero " << numero << " tiene " << contador_recursivo(
            numero) << " digitos " << endl;
18     else
19         cout << "El numero debe ser positivo." << endl;
20     return 0;
21 }
```

Programa 4.37: Lenguaje C \Leftarrow Ejercicio 73

```
1 def contador_recursivo(numero):
2     if (numero == 0):
3         return 0
4     else:
5         return 1 + contador_recursivo(int(numero/10))
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese el numero : "))
9     if (numero >= 0):
10         print(str(numero)+" tiene "+str(contador_recursivo(numero))+
            " digitos")
11     else:
12         print("Error el numero debe ser positivo")
```

Programa 4.38: Lenguaje Python \Leftarrow Ejercicio 73

Ejercicio 74. Escribir un programa que implemente el método recursivo que devuelva la suma de los números desde n hasta m , ambos incluidos con $m \geq n$, si $n = m$ se debe retornar el valor de n .

```
1 #include <iostream>
2 using namespace std;
3
4 int sumador_recursivo(int n, int m){
5     if(m == n)
6         return n;
7     else
8         return n + sumador_recursivo(n + 1, m);
9 }
10
```

```
11
12 int main() {
13     int numero1, numero2;
14     cout << "ingrese un numero entero : ";
15     cin >> numero1;
16     cout << "ingrese otro numero entero : ";
17     cin >> numero2;
18     if (numero1 <= numero2)
19         cout << "los numeros entre " << numero1 << " y " << numero2 << "
            suman " << sumador_recursivo(numero1, numero2) << endl;
20     else
21         cout << "Error los numeros deben ser ingresados de menor a mayor."
            << endl;
22     return 0;
23 }
```

Programa 4.39: Lenguaje C \Leftarrow Ejercicio 74

```
1 def sumador_recursivo(n,m):
2     if (n == m):
3         return n
4     else:
5         return n + sumador_recursivo(n+1, m)
6
7 if __name__ == "__main__":
8     n = int(input("ingrese el primer numero : "))
9     m = int(input("ingrese el segundo numero : "))
10    if (m >= n):
11        print("los numeros entre " + str(n) + " y " + str(m) + " suman "
            + str(sumador_recursivo(n,m)))
12    else:
13        print("Error los numeros deben ser ingresados de menor a mayor."
            )
```

Programa 4.40: Lenguaje Python \Leftarrow Ejercicio 74

Ejercicio 75. Escribir un programa que usando recursión valide si un dígito n pertenece a un número m

```
1 #include <iostream>
2 using namespace std;
3
4 bool validador_recursivo(int m, int n){
5     if(m == 0)
6         return false;
7     if(m%10 == n)
8         return true;
9     else
10        return validador_recursivo(m/10,n);
11 }
12
13
14 int main() {
15     int numero, digito;
16     cout << "ingrese un numero entero : ";
17     cin >> numero;
```

```
18 cout << "ingrese el digito a verificar : ";
19 cin >> digito;
20 if (validador_recursivo(numero, digito))
21     cout << "el digito " << digito << " se encuentra en " << numero <<
        endl;
22 else
23     cout << "el digito " << digito << " no se encuentra en " << numero
        << endl;
24 return 0;
25 }
```

Programa 4.41: Lenguaje C \Leftarrow Ejercicio 75

```
1 def validador_recursivo(m, n):
2     if(m == 0):
3         return False
4     if(m%10 == n):
5         return True
6     else:
7         return validador_recursivo(int(m/10),n);
8
9 if __name__ == "__main__":
10     numero = int(input("ingrese un numero : "))
11     digito = int(input("ingrese el digito a verificar : "))
12     if (validador_recursivo(numero, digito)):
13         print("el digito " + str(digito) + " se encuentra en " + str(
            numero))
14     else:
15         print("el digito " + str(digito) + " no se encuentra en " + str(
            numero))
```

Programa 4.42: Lenguaje Python \Leftarrow Ejercicio 75

Ejercicio 76. Escribir un programa que mediante recursividad permita invertir un número entero positivo leído desde el teclado.

```
1 #include <iostream>
2 #include <cstring>
3
4 using namespace std;
5 string str(int n){
6     string rtn="";
7     while(n>0){
8         rtn.insert(rtn.begin(), n%10+'0');
9         n/=10;
10    }
11    return rtn;
12 }
13
14 string inversor_recursivo(int numero){
15     if(numero == 0)
16         return "";
17     else
18         return str(numero % 10) + (inversor_recursivo(numero/10)) ;
19 }
20
```

```
21
22 int main() {
23     int numero;
24     cout << "ingrese un numero entero : ";
25     cin >> numero;
26
27     if (numero >= 0)
28         cout << inversor_recursivo(numero) << endl;
29     else
30         cout << "Error el numero debe ser positivo " << endl;
31     return 0;
32 }
```

Programa 4.43: Lenguaje C \Leftarrow Ejercicio 76

```
1 def inversor_recursivo(numero):
2     if(numero == 0):
3         return ""
4     else:
5         return str(numero % 10) + (inversor_recursivo(int(numero/10)))
6
7 if __name__ == "__main__":
8     numero = int(input("ingrese un numero : "))
9     if (numero >= 0):
10         print(inversor_recursivo(numero))
11     else:
12         print("Error el numero debe ser positivo ");
```

Programa 4.44: Lenguaje Python \Leftarrow Ejercicio 76

Ejercicio 77. Escribir un programa que permita determinar si un número es impar utilizando recursividad mutua o cruzada en la que x es par si $x-1$ es impar y x es impar si $x-1$ es par donde se sabe que 0 es par.

```
1 #include <iostream>
2 using namespace std;
3
4 bool par(int);
5 bool impar(int);
6
7 int main() {
8     int numero;
9     cout << "ingrese un numero entero : ";
10    cin >> numero;
11
12    if (impar(numero))
13        cout << numero << " es impar" << endl;
14    else
15        cout << numero << " es par" << endl;
16    return 0;
17 }
18
19 bool par(int n){
20     if(n == 0)
```

```
21     return true;
22 else
23     return impar(n-1);
24 }
25
26 bool impar(int n){
27     if(n == 0)
28         return false;
29     else
30         return par(n-1);
31 }
```

Programa 4.45: Lenguaje C \Leftarrow Ejercicio 77

```
1 def par(n):
2     if(n == 0):
3         return True
4     else:
5         return impar(n-1)
6
7 def impar(n):
8     if(n == 0):
9         return False
10    else:
11        return par(n-1)
12
13 if __name__ == "__main__":
14     numero = int(input("ingrese un numero : "))
15     if (impar(numero)):
16         print(str(numero) + " es impar")
17     else:
18         print(str(numero) + " es par")
```

Programa 4.46: Lenguaje Python \Leftarrow Ejercicio 77

Capítulo 5

Estructuras de Datos

5.1. Arreglos

Ejercicio 78. Escribir un programa que dado un arreglo de las vocales y otro con las consonantes, capture una cadena de caracteres y cuente cuantos caracteres pertenecen al arreglo de vocales, cuantos al arreglo de consonantes y cuantos no pertenecen a ninguno de los dos.

```
1 #include <iostream>
2 #include <cstring>
3 using namespace std;
4
5 int contar_letras(string cad, string letras){
6     int cont = 0;
7     for(int c = 0; c < cad.size(); c++){
8         for(int l = 0; l < letras.size(); l++){
9             if(cad.substr(c, 1) == letras.substr(l,1))
10                 cont++;
11         }
12     }
13     return cont;
14 }
15
16 int main(int argc, char *argv[]) {
17     string vocales = "aeiou";
18     string consonantes = "bcdfghijklmnpqrstvwxyz";
19     string cadena;
20     int cont_vocales, cont_consonantes, cont_otros;
21
22     cout << "ingrese una cadena : " << endl;
23     getline(cin, cadena);
24
25     cont_vocales = contar_letras(cadena, vocales);
26     cont_consonantes = contar_letras(cadena, consonantes);
27     cont_otros = cadena.size() - (cont_vocales + cont_consonantes);
28
29     cout << "la cantidad de vocales es: " << cont_vocales << endl;
30     cout << "la cantidad de consonantes es: " << cont_consonantes <<
```

```
endl;
31 cout << "la cantidad de otros caracteres es: " << cont_otros << endl
;
32
33 return 0;
34 }
```

Programa 5.1: Lenguaje C \Leftarrow Ejercicio 78

```
1 def contar_letras(cad, let):
2     contador = 0
3     for c in cad:
4         for l in let:
5             if c == l:
6                 contador = contador + 1
7     return contador
8
9 vocales = list("aeiou")
10 consonantes = list("bcdfghijklmnpqrstvwxyz")
11 cadena = list(str(input("ingrese la cadena a evaluar: ")))
12 contador_vocales = contar_letras(cadena, vocales)
13 contador_consonantes = contar_letras(cadena, consonantes)
14 contador_otros = len(cadena) - (contador_vocales +
    contador_consonantes)
15 print("la cantidad de vocales es: " + str(contador_vocales))
16 print("la cantidad de consonantes es: " + str(contador_consonantes))
17 print("la cantidad de otros caracteres es: " + str(contador_otros))
```

Programa 5.2: Lenguaje Python \Leftarrow Ejercicio 78

Ejercicio 79. Escribir un programa que usando arreglos de dígitos validos verifique si un número entero positivo ingresado por el usuario es válido en un sistema con base X (2,8,10) dada también por el usuario.

```
1 #include<iostream>
2 using namespace std;
3
4 int main() {
5
6     int base;
7     int base10[10];
8     int base2[2];
9     int base8[8];
10    int digito;
11    bool estaen;
12    int n;
13    int numero;
14
15    cout << "Ingrese un numero entero: ";
16    cin >> numero;
17    cout << "Ingrese la base del numero: ";
18    cin >> base;
19
20    if (numero >= 0) {
21        for (int i = 0; i <= 1; i++) {
```

```
22     base2[i] = i;
23 }
24
25 for (int i = 0; i <= 7; i++) {
26     base8[i] = i;
27 }
28
29 for ( int i = 0; i <= 9; i++) {
30     base10[i] = i;
31 }
32
33 n = numero;
34
35 switch ( base ) {
36 case 2:
37     while ( n > 0 ) {
38         digito = n % 10;
39         estaen = false;
40         for (int i = 0; i <= 1; i++) {
41             if (base2[i] == digito) {
42                 estaen = true;
43                 i = 2;
44             }
45         }
46         n = n / 10 ;
47         if ( !estaen ) {
48             n = -1;
49         }
50     }
51     if (n == 0) {
52         cout << numero << " esta en Base 2" << endl;
53     } else {
54         cout << numero << " No esta en Base 2" << endl;
55     }
56     break;
57 case 8:
58     while ( n>0 ) {
59         digito = n % 10;
60         estaen = false;
61         for (int i = 0; i <= 7; i++) {
62             if (base8[i] == digito) {
63                 estaen = true;
64                 i = 8;
65             }
66         }
67         n = n / 10;
68         if ( !estaen ) {
69             n = -1;
70         }
71     }
72     if (n == 0) {
73         cout << numero << " esta en Base 8" << endl;
74     } else {
75         cout << numero << " No esta en Base 8" << endl;
76     }
77     break;
78 case 10:
79     while (n > 0) {
80         digito = n % 10;
```

```
81     estaen = false;
82     for (int i = 0; i <= 9; i++) {
83         if (base10[i] == digito) {
84             estaen = true;
85             i = 10;
86         }
87     }
88     n = n/10 ;
89     if ( !estaen ) {
90         n -= 1;
91     }
92 }
93 if (n == 0) {
94     cout << numero << " esta en Base 10" << endl;
95 } else {
96     cout << numero << " No esta en Base 10" << endl;
97 }
98 break;
99 default:
100     cout << "La base " << base << " no esta definida" << endl;
101     break;
102 }
103 } else {
104     cout << "Error en el ingreso de datos, " << numero << " debe ser
105     >= 0" << endl;
106 }
107 return 0;
108
109 }
```

Programa 5.3: Lenguaje C \Leftarrow Ejercicio 79

```
1 numero = int(input("Ingrese un numero entero: "))
2 base = int(input("Ingrese la base del numero: "))
3
4 base2 = []
5 base8 = []
6 base10 = []
7
8 if numero >= 0:
9     for i in range(0,2):
10         base2.append(i)
11
12     for i in range(0,8):
13         base8.append(i)
14
15     for i in range(0,10):
16         base10.append(i)
17
18     n = numero
19
20     if base == 2:
21         while (n > 0):
22             digito = n % 10
23             estaen = False
24             if digito in base2:
25                 estaen = True
26
```



```
27         n = n // 10
28
29         if not estaen:
30             n = -1
31
32     if n == 0:
33         print(str(numero) + " esta en Base 2")
34     else:
35         print(str(numero) + " No esta en Base 2")
36 elif base == 8:
37     while (n > 0):
38         digito = n % 10
39         estaen = False
40         if digito in base8:
41             estaen = True
42         n = n // 10
43
44         if not estaen:
45             n = -1
46
47     if n == 0:
48         print(str(numero) + " esta en Base 8")
49     else:
50         print(str(numero) + " No esta en Base 8")
51 elif base == 10:
52     while (n > 0):
53         digito = n % 10
54         estaen = False
55         if digito in base10:
56             estaen = True
57         n = n // 10
58
59         if not estaen:
60             n = -1
61
62     if n == 0:
63         print(str(numero) + " esta en Base 10")
64     else:
65         print(str(numero) + " No esta en Base 10")
66 else:
67     print("La base " + str(base) + " no esta definida")
68 else:
69     print("Error en el ingreso de datos , " + str(numero) + " debe ser
    >= 0")
```

Programa 5.4: Lenguaje Python \Leftarrow Ejercicio 79

Ejercicio 80. Escribir un programa que tras asignar números negativos y positivos de manera aleatoria a un arreglo de 10 posiciones calcule, independientemente, la suma de los elementos positivos y negativos.

```
1 #include <iostream>
2 #include <ctime>
3 #include <cstdlib>
4 using namespace std;
```

```
5
6 int main(){
7     int ar[10];
8     int cont_positivos = 0, cont_negativos = 0;
9     srand(time(0));
10
11     for(int i = 0; i < 10; i++){
12         ar[i] = (rand() % 100) - 50;
13     }
14
15     for(int i = 0; i < 10; i++){
16         if(ar[i] > 0)
17             cont_positivos += ar[i];
18         else
19             cont_negativos += ar[i];
20     }
21
22     cout << "Elementos del arreglo: " << endl;
23     for(int i=0;i<10;i++)
24         cout << ar[i] << " ";
25
26     cout << endl;
27
28     cout << " los positivos suman : " << cont_positivos << endl;
29     cout << " los negativos suman : " << cont_negativos << endl;
30
31     return 0;
32 }
```

Programa 5.5: Lenguaje C \Leftarrow Ejercicio 80

```
1 import random
2
3 lista = []
4 for i in range(10):
5     lista.append(random.randint(-50, 50))
6
7 positivos = 0
8 negativos = 0
9
10 for e in lista:
11     if e >= 0:
12         positivos = positivos + e
13     else:
14         negativos = negativos + e
15
16
17 print(lista)
18 print("la suma de los positivos es: " + str(positivos))
19 print("la suma de los negativos es: " + str(negativos))
```

Programa 5.6: Lenguaje Python \Leftarrow Ejercicio 80

Ejercicio 81. Escribir un programa que tras asignar 10 números de manera aleatoria a un arreglo, determine las posiciones de este en las

que se encuentran el máximo, el mínimo valor y el valor más cercano al promedio entero.

```
1 #include <iostream>
2 #include <ctime>
3 #include <cstdlib>
4 using namespace std;
5
6 int main(){
7     int ar[10];
8     int max, min, cerca, promedio, acum = 0;
9     srand(time(0));
10    for(int i = 0; i < 10; i++){
11        ar[i] = (rand() % 100);
12        acum = acum + ar[i];
13    }
14    promedio = acum / 10;
15    max = ar[0];
16    min = ar[0];
17    cerca = ar[0];
18    for(int i = 0; i < 10; i++){
19        if(ar[i] > max){
20            max = ar[i];
21        }
22        if(ar[i] < min){
23            min = ar[i];
24        }
25        if(abs(promedio - ar[i]) < abs(promedio - cerca)){
26            cerca = ar[i];
27        }
28    }
29
30    cout << "Elementos del arreglo: " << endl;
31    for(int i=0;i<10;i++)
32        cout << ar[i] << " ";
33
34    cout << endl << " el menor es : " << min << endl;
35    cout << " el mayor es : " << max << endl;
36    cout << " el promedio es: " << promedio << endl;
37    cout << " el mas cercano a " << promedio << " es : " << cerca << endl;
38    ;
39    return 0;
40 }
```

Programa 5.7: Lenguaje C \Leftarrow Ejercicio 81

```
1 import random
2 import array
3
4 ar = []
5 #ar es una lista
6 acum = 0
7
8 for i in range(0,10):
9     ar.append(random.randint(0, 100))
10    acum += ar[i]
11
```

```
12 promedio = acum // 10
13 maximo = ar[0]
14 minimo = ar[0]
15 cerca = ar[0]
16
17 for i in range(0,10):
18     if ar[i] > maximo:
19         maximo = ar[i]
20     if ar[i] < minimo:
21         minimo = ar[i]
22     if abs(promedio - ar[i]) < abs(promedio - cerca):
23         cerca = ar[i]
24
25 print("Elementos del arreglo:\n" + str(ar))
26
27 print( " el menor es : " + str(minimo) )
28 print( " el mayor es : " + str(maximo) )
29 print( " el promedio es: " + str(promedio) )
30 print( " el mas cercano a "+ str(promedio) + " es : " + str(cerca) )
```

Programa 5.8: Lenguaje Python \Leftarrow Ejercicio 81

Ejercicio 82. Escribir un programa que inicialice un arreglo con números aleatorios enteros entre 0 y 300 y calcule por medio de un método recursivo el elemento mayor.

```
1 #include <iostream>
2 #include <stdlib.h>
3 #include <time.h>
4 using namespace std;
5
6 /*
7  Imprime los elementos del arreglo desde la posición i hasta t - 1
8  precondiciones
9  0 <= i < t
10 */
11 void imprimir(int a[], int i, int t){
12
13     if(i < t ){
14         cout << a[i] << " ";
15         imprimir (a,i+1,t);
16     }
17 }
18
19
20 /*
21  Inicializa el arreglo desde la posición t hasta la posición 0, con
22  valores enteros aleatorios entre 0 y 300.
23  precondiciones
24  0 <= t < tamaño del arreglo
25 */
26 void inicializar(int a[], int t){
27     if (t >= 0 ){
28         a[t] = rand() % 300;
29         inicializar(a,t-1);
30     }
31 }
```

```
30 }
31
32 /*
33 Determina el mayor de dos números enteros
34 */
35 int mayor(int a, int b){
36     return a < b ? b : a;
37 }
38
39
40 /*
41 Encuentra el elemento mayor almacenado en un arreglo de enteros, desde
    la posición t hasta la posición 0
42 precondiciones
43 0 <= t < tamaño del arreglo
44 */
45 int mayor (int a[], int t)
46 {
47     if (t==0)
48         return a[0];
49     else
50
51         return mayor(a[t], mayor(a, t-1));
52 }
53 /*
54 Función principal.
55 */
56 int main(){
57     srand(time(0));
58     int n;
59     cout << "Dígite la dimensión del arreglo: " ;
60     cin >> n;
61     int datos[n];
62     inicializar(datos,n-1);
63     cout << endl;
64     imprimir(datos,0,n);
65     cout << endl;
66     cout << "el mayor elemento del arreglo es: " << mayor(datos,n-1) <<
        endl;
67     return 0;
68 }
```

Programa 5.9: Lenguaje C \Leftarrow Ejercicio 82

```
1 import random
2 import array
3
4 def inicializar(datos, n):
5     if n >= 0:
6         datos.append(random.randint(0, 100))
7         inicializar(datos, n-1)
8
9 def mayor(a, b):
10     if a < b:
11         return b
12     else:
13         return a
14
15 def mayorE(datos,n):
```

```
16     if n == 0:
17         return datos[0]
18     else:
19         return mayor(datos[n], mayorE(datos, n-1))
20
21 if __name__ == "__main__":
22     n = int(input("Dígite la dimension del arreglo: "))
23     datos = []
24     inicializar(datos, n-1)
25     print(datos)
26     print("el mayor elemento del arreglo es: " + str(mayorE(datos, n-1)
    ))
```

Programa 5.10: Lenguaje Python \Leftarrow Ejercicio 82

Ejercicio 83. Escribir un programa que usando un arreglos de enteros de 10 posiciones, inicializado con numeros aleatorios entre 100 y 200 permita ordenar dicho arreglo usando un método de ordenamiento de burbuja.

```
1 #include <iostream>
2 #include <ctime>
3 #include <cstdlib>
4 using namespace std;
5
6 int main(){
7     int ar[10];
8     int temp;
9     srand(time(0));
10
11     for(int i = 0; i < 10; i++){
12         ar[i] = (rand() % 100) + 100;
13     }
14
15     cout << "Elementos del arreglo original: " << endl;
16     for(int i=0;i<10;i++)
17         cout << ar[i] << " ";
18
19     cout << endl;
20
21     for(int i = 0; i < 10; i++){
22         for(int j = i; j < 10; j++){
23             if(ar[i] > ar[j]){
24                 temp = ar[i];
25                 ar[i] = ar[j];
26                 ar[j] = temp;
27             }
28         }
29     }
30
31     cout << "Elementos del arreglo ordenado: " << endl;
32     for(int i=0;i<10;i++)
33         cout << ar[i] << " ";
34
35     cout << endl;
36
```

```
37 return 0;
38 }
```

Programa 5.11: Lenguaje C \Leftarrow Ejercicio 83

```
1 import random
2
3 lista = []
4 for i in range(10):
5     lista.append(random.randint(100, 200))
6
7 print("Elementos del arreglo original: ")
8 print(lista)
9
10 for i in range(10):
11     for j in range(i,10):
12         if lista[i] > lista[j]:
13             temp = lista[i]
14             lista[i] = lista[j]
15             lista[j] = temp
16 print()
17
18 print("Elementos del arreglo ordenado: ")
19 print(lista)
```

Programa 5.12: Lenguaje Python \Leftarrow Ejercicio 83

5.2. Matrices

Ejercicio 84. Escribir un programa que determine la posición de una matriz de 3×3 en la que se encuentra el valor máximo de una serie de números generados aleatoriamente.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5 //procedimiento que llena aleatoriamente la matriz
6 void llenado_aleatorio(int [3][3]);
7
8 //procedimiento que muestra en pantalla la matriz
9 void ver_matriz(int [3][3]);
10
11 int main(int argc, char *argv[]) {
12     int matriz[3][3];
13     int mayor, fila_mayor = 0, columna_mayor = 0;
14
15     srand(time(0));
16     llenado_aleatorio(matriz);
17
18     ver_matriz(matriz);
19
20     mayor = matriz[fila_mayor][columna_mayor];
21     for(int f=0; f<3; f++){
```

```
22     for(int c=0; c<3; c++){
23         if(mayor<matriz[f][c]){
24             fila_mayor = f;
25             columna_mayor = c;
26             mayor=matriz[fila_mayor][columna_mayor];
27         }
28     }
29 }
30
31 ver_matriz(matriz);
32
33 cout << "mayor = " << mayor << endl;
34 cout << "en la posicion = (" << fila_mayor << ", " << columna_mayor
<< ")"<< endl;
35
36 return 0;
37 }
38
39 void llenado_aleatorio(int m[3][3]){
40     for(int f=0; f<3; f++){
41         for(int c=0; c<3; c++){
42             m[f][c] = rand() % 100;
43         }
44     }
45 }
46 void ver_matriz(int m[3][3]){
47     for(int f=0; f<3; f++){
48         for(int c=0; c<3; c++){
49             cout << m[f][c] << "\t";
50         }
51         cout << endl;
52     }
53 }
```

Programa 5.13: Lenguaje C \Leftarrow Ejercicio 84

```
1 from random import randint
2 def llenado_aleatorio(m):
3     for i in range(3):
4         m.append([randint(0, 100) for i in range(3)])
5
6 def mostrar_matriz(m):
7     for i in m:
8         for j in i:
9             print(j, end='\t')
10        print("")
11
12 def main():
13     matriz=[]
14     fila_mayor = 0
15     columna_mayor = 0
16
17     llenado_aleatorio(matriz)
18     mostrar_matriz(matriz)
19
20     mayor=matriz[fila_mayor][columna_mayor]
21
22     for f in range(3):
23         for c in range(3):
```



```
24         if mayor < matriz[f][c]:
25             fila_mayor = f
26             columna_mayor = c
27             mayor = matriz[fila_mayor][columna_mayor]
28
29     print("mayor = " + str(mayor))
30     print("en la posicion = (" + str(fila_mayor) + "," + str(
31         columna_mayor) + ")")
32 if __name__ == "__main__":
33     main()
```

Programa 5.14: Lenguaje Python \Leftarrow Ejercicio 84

Ejercicio 85. Escribir un programa que sume, independientemente, los elementos positivos y negativos de la una matriz de 5×5 , la matriz se debe llenar solicitando los datos al usuario.

```
1 #include <iostream>
2 using namespace std;
3 //procedimiento que llena la matriz
4 void llenado(int [5][5]);
5
6 //procedimiento que muestra en pantalla la matriz
7 void ver_matriz(int [5][5]);
8
9 int main(int argc, char *argv[]) {
10     int matriz[5][5];
11     int positivos=0;
12     int negativos=0;
13
14     llenado(matriz);
15
16     for(int f=0; f<5; f++){
17         for(int c=0; c<5; c++){
18             if(matriz[f][c]>0){
19                 positivos+=matriz[f][c];
20             }else{
21                 negativos+=matriz[f][c];
22             }
23         }
24     }
25
26     ver_matriz(matriz);
27
28     cout << "positivos = " << positivos << endl;
29     cout << "negativos = " << negativos << endl;
30
31     return 0;
32 }
33
34 void llenado(int m[5][5]){
35     for(int f=0; f<5; f++){
36         for(int c=0; c<5; c++){
37             cout<<"ingrese el valor para la fila "<<f<<" y la columna "<<c<<
38                 " : ";
```

```
38     cin>>m[f][c];
39 }
40 }
41 }
42
43 void ver_matriz(int m[5][5]){
44     for(int f=0; f<5; f++){
45         for(int c=0; c<5; c++){
46             cout << m[f][c] << "\t";
47         }
48         cout << endl;
49     }
50 }
```

Programa 5.15: Lenguaje C \Leftarrow Ejercicio 85

```
1 __size = 5
2
3 def llenado_usuario(m):
4     for i in range(__size):
5         n=[]
6         for j in range(__size):
7             num = int(input("ingrese el valor para la posicion (" +
8                             str(i) + "," + str(j) + ") : "))
9             n.append(num)
10            m.append(n)
11
12 def mostrar_matriz(m):
13     for i in m:
14         for j in i:
15             print(j, end='\t')
16         print("")
17
18 if __name__ == "__main__":
19     matriz=[]
20     positivos = 0
21     negativos = 0
22     llenado_usuario(matriz)
23     mostrar_matriz(matriz)
24     for i in matriz:
25         for j in i:
26             if j > 0:
27                 positivos += j
28             else:
29                 negativos += j
30
31     print("positivos = " + str(positivos))
32     print("negativos = " + str(negativos))
```

Programa 5.16: Lenguaje Python \Leftarrow Ejercicio 85

Ejercicio 86. Escribir un programa que llene la primera fila de una matriz de 3 filas por 10 columnas con números aleatorios entre 1 y 20, la segunda fila con los cuadrados de los datos de la primera fila y la tercera fila con la suma de la primera y la segunda.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5 //procedimiento que llena aleatoriamente la primera fila de la matriz
6 void llenado_aleatorio(int [3][10]);
7
8 //procedimiento que llena con cuadrados de la primera fila de la
  matriz
9 //la segunda fila
10 void llenado_cuadrado(int [3][10]);
11
12 //procedimiento que llena con la suma de la primera y segunda fila de
  la matriz
13 //la tercera fila
14 void llenado_sumas(int [3][10]);
15
16 //procedimiento que muestra en pantalla la matriz
17 void ver_matriz(int [3][10]);
18
19 int main(int argc, char *argv[]) {
20     int matriz[3][10];
21
22     srand(time(0));
23     llenado_aleatorio(matriz);
24     llenado_cuadrado(matriz);
25     llenado_sumas(matriz);
26
27     ver_matriz(matriz);
28
29     return 0;
30 }
31
32 void llenado_aleatorio(int m[3][10]){
33     for(int c=0; c<10; c++){
34         m[0][c] = 1 + rand() % 19;
35     }
36 }
37
38 void llenado_cuadrado(int m[3][10]){
39     for(int c=0; c<10; c++){
40         m[1][c] = m[0][c] * m[0][c];
41     }
42 }
43
44 void llenado_sumas(int m[3][10]){
45     for(int c=0; c<10; c++){
46         m[2][c] = m[0][c] + m[1][c];
47     }
48 }
49
50 void ver_matriz(int m[3][10]){
51     for(int f=0; f<3; f++){
52         for(int c=0; c<10; c++){
53             cout << m[f][c] << "\t";
54         }
55         cout << endl;
56     }
```

57 }

Programa 5.17: Lenguaje C \Leftarrow Ejercicio 86

```
1 from random import randint
2 __columnas = 10
3
4 def llenado_aleatorio(m):
5     n=[]
6     for i in range(__columnas):
7         n.append(randint(1, 20))
8     m.append(n)
9
10 def llenado_cuadrados(m):
11     n=[]
12     for i in range(__columnas):
13         n.append(m[0][i]**2)
14     m.append(n)
15
16 def llenado_sumas(m):
17     n=[]
18     for i in range(__columnas):
19         n.append(m[0][i]+m[1][i])
20     m.append(n)
21
22 def mostrar_matriz(m):
23     for i in m:
24         for j in i:
25             print(j, end='\t')
26         print("")
27
28 if __name__ == "__main__":
29     matriz=[]
30     llenado_aleatorio(matriz)
31     llenado_cuadrados(matriz)
32     llenado_sumas(matriz)
33     mostrar_matriz(matriz)
```

Programa 5.18: Lenguaje Python \Leftarrow Ejercicio 86

Ejercicio 87. Escribir un programa que divida todos los elementos de una matriz $M(3,4)$ por el elemento situado en la posición $(2,2)$.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5 //procedimiento que llena aleatoriamente la matriz
6 void llenado_aleatorio(int [3][4]);
7
8 //procedimiento que muestra en pantalla la matriz
9 void ver_matriz(int [3][4]);
10
11 int main(int argc, char *argv[]) {
12     int matriz[3][4];
13     int divisor;
```

```
14
15 srand(time(0));
16 llenado_aleatorio(matriz);
17
18 cout << "matriz original" << endl;
19 ver_matriz(matriz);
20
21 divisor = matriz[2][2];
22 for(int f=0; f<3; f++){
23     for(int c=0; c<4; c++){
24         matriz[f][c]/=divisor;
25     }
26 }
27
28 cout << "matriz dividida" << endl;
29 ver_matriz(matriz);
30
31 return 0;
32 }
33
34 void llenado_aleatorio(int m[3][4]){
35     for(int f=0; f<3; f++){
36         for(int c=0; c<4; c++){
37             m[f][c] = 1 + rand() % 9;
38         }
39         cout << endl;
40     }
41 }
42 void ver_matriz(int m[3][4]){
43     for(int f=0; f<3; f++){
44         for(int c=0; c<4; c++){
45             cout << m[f][c] << "\t";
46         }
47         cout << endl;
48     }
49 }
```

Programa 5.19: Lenguaje C \Leftarrow Ejercicio 87

```
1 from random import randint
2
3 def llenado_aleatorio(m):
4     for i in range(3):
5         m.append([randint(0, 100) for i in range(4)])
6
7 def mostrar_matriz(m):
8     for i in m:
9         for j in i:
10             print(j, end='\\t')
11         print("")
12
13 if __name__ == "__main__":
14     matriz=[]
15     llenado_aleatorio(matriz)
16     print("matriz original")
17     mostrar_matriz(matriz)
18
19     divisor = matriz[2][2]
20     for i in range(3):
```

```
21         for j in range(4):
22             matriz[i][j] = int(matriz[i][j] / divisor)
23
24     print("matriz dividida")
25     mostrar_matriz(matriz)
```

Programa 5.20: Lenguaje Python \Leftarrow Ejercicio 87

Ejercicio 88. Escribir un programa que permita ordenar los elementos de una matriz de $M(5, 5)$ usando el método de burbuja, el programa debe inicializar la matriz con números aleatorios multiples de 5 comprendidos entre 1 y 100 y mostrar la matriz en pantalla de forma ordenada tabulando a derecha cada uno de los elementos de la matriz.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5
6 //procedimiento que llena aleatoriamente la matriz
7 void llenado_aleatorio(int m[5][5]);
8
9 //procedimiento que muestra en pantalla la matriz
10 void ver_matriz(int m[5][5]);
11
12 //procedimiento que ordena la matriz
13 void ordenar_matriz(int m[5][5]);
14
15 int main(int argc, char *argv[]) {
16     int matriz[5][5];
17
18     srand(time(0));
19     llenado_aleatorio(matriz);
20
21     cout << "matriz original" << endl;
22     ver_matriz(matriz);
23
24     ordenar_matriz(matriz);
25
26     cout << "matriz ordenada" << endl;
27     ver_matriz(matriz);
28     return 0;
29 }
30
31 void llenado_aleatorio(int m[5][5]){
32     int aleatorio;
33     for(int f=0; f<5; f++){
34         for(int c=0; c<5; c++){
35             do{
36                 aleatorio = 1 + rand() % 99;
37             }while(aleatorio % 5 != 0);
38             m[f][c] = aleatorio;
39         }
40         cout << endl;
41     }
```

```
42 }
43 void ver_matriz(int m[5][5]){
44     for(int f=0; f<5; f++){
45         for(int c=0; c<5; c++){
46             if(m[f][c] < 10)
47                 cout << " " << m[f][c] << "\t";
48             else
49                 cout << m[f][c] << "\t";
50         }
51         cout << endl;
52     }
53 }
54 void ordenar_matriz(int m[5][5]){
55     int temp;
56     for(int f=0; f<5; f++){
57         for(int c=0; c<5; c++){
58             for(int fa=0; fa<5; fa++){
59                 for(int ca=0; ca<5; ca++){
60                     if(m[f][c] < m[fa][ca]){
61                         temp = m[f][c];
62                         m[f][c] = m[fa][ca];
63                         m[fa][ca] = temp;
64                     }
65                 }
66             }
67         }
68     }
69 }
```

Programa 5.21: Lenguaje C \Leftarrow Ejercicio 88

```
1 from random import randint
2
3 __size = 5
4
5 def llenado_aleatorio(m):
6     for i in range(__size):
7         m.append([randint(1, 100) for i in range(__size)])
8
9 def mostrar_matriz(m):
10    for i in m:
11        for j in i:
12            if(j < 10):
13                print(" " + str(j), end='\t')
14            else:
15                print(" " + str(j), end='\t')
16        print("")
17
18 def ordenar_matriz(m):
19     for i in range(__size):
20         for j in range(__size):
21             for ia in range(__size):
22                 for ja in range(__size):
23                     if(m[i][j] < m[ia][ja]):
24                         temp = m[i][j]
25                         m[i][j] = m[ia][ja]
26                         m[ia][ja] = temp
27
28 if __name__ == "__main__":
```

```
29     matriz=[]
30     llenado_aleatorio(matriz)
31     print("matriz original")
32     mostrar_matriz(matriz)
33     ordenar_matriz(matriz)
34     print("matriz ordenada")
35     mostrar_matriz(matriz)
```

Programa 5.22: Lenguaje Python \Leftarrow Ejercicio 88

Ejercicio 89. Dada una matriz cuadrada de 4 por 4 inicializada con números aleatorios comprendidos entre 10 y 50, escribir un programa que genere una matriz que al sumarla elemento a elemento con la matriz original convierta en 1 los números de la matriz del triangulo superior a la diagonal principal de la matriz y en -1 los números del triangulo inferior a la diagonal principal, los números de la diagonal principal deben convertirse a 0. Se debe mostrar en pantalla la matriz original, la matriz generada para convertir la original y la matriz convertida.

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5
6 //procedimiento que llena aleatoriamente la matriz
7 void llenado_aleatorio(int [4][4]);
8
9 //procedimiento que muestra en pantalla la matriz
10 void ver_matriz(int [4][4]);
11
12 //procedimiento que genera la diagonal principal
13 void generar_diagonal(int [4][4],int [4][4]);
14
15 //procedimiento que genera la el triangulo superior
16 void generar_superior(int [4][4],int [4][4]);
17
18 //procedimiento que genera la el triangulo inferior
19 void generar_inferior(int [4][4],int [4][4]);
20
21 //procedimiento que genera la el triangulo inferior
22 void sumar_matrices(int [4][4],int [4][4]);
23
24 int main(int argc, char *argv[]) {
25     int matriz[4][4];
26     int auxiliar[4][4];
27
28     srand(time(0));
29     llenado_aleatorio(matriz);
30
31     cout << "matriz original" << endl;
32     ver_matriz(matriz);
33
34     generar_diagonal(matriz,auxiliar);
35     generar_superior(matriz,auxiliar);
```



```
36  generar_inferior(matriz , auxiliar);
37
38  sumar_matrices(matriz , auxiliar);
39
40  cout << "matriz gererada" << endl;
41  ver_matriz(auxiliar);
42
43  cout << "matriz resultado" << endl;
44  ver_matriz(matriz);
45  return 0;
46 }
47
48 void llenado_aleatorio(int m[4][4]){
49     int aleatorio;
50     for(int f=0; f<4; f++){
51         for(int c=0; c<4; c++){
52             aleatorio = 10 + rand() % 40;
53             m[f][c] = aleatorio;
54         }
55         cout << endl;
56     }
57 }
58 void ver_matriz(int m[4][4]){
59     for(int f=0; f<4; f++){
60         for(int c=0; c<4; c++){
61             if(m[f][c] >= 0)
62                 cout << " " << m[f][c] << "\t";
63             else
64                 cout << m[f][c] << "\t";
65         }
66         cout << endl;
67     }
68 }
69
70 void generar_diagonal(int m[4][4],int a[4][4]){
71     for(int f=0; f<4; f++){
72         a[f][f] = m[f][f] * -1;
73     }
74 }
75
76 void generar_superior(int m[4][4],int a[4][4]){
77     for(int f=0; f<4; f++){
78         for(int c=0; c<4; c++){
79             if(c > f)
80                 a[f][c] = (m[f][c] * -1) + 1;
81         }
82     }
83 }
84
85 void generar_inferior(int m[4][4],int a[4][4]){
86     for(int f=0; f<4; f++){
87         for(int c=0; c<4; c++){
88             if(c < f)
89                 a[f][c] = (m[f][c] * -1) - 1;
90         }
91     }
92 }
93
94 void sumar_matrices(int m[4][4],int a[4][4]){
```

```
95  for(int f=0; f<4; f++){
96      for(int c=0; c<4; c++){
97          m[f][c] = m[f][c] + a[f][c];
98      }
99  }
100 }
```

Programa 5.23: Lenguaje C \Leftarrow Ejercicio 89

```
1 from random import randint
2
3 __size = 4
4
5 def llenado_aleatorio(m):
6     for i in range(__size):
7         m.append([randint(10, 50) for i in range(__size)])
8
9 def mostrar_matriz(m):
10     for i in m:
11         for j in i:
12             if j < 0:
13                 print(str(j), end='\t')
14             else:
15                 print(" " + str(j), end='\t')
16         print("")
17
18 def generar_auxiliar(m, a):
19     for i in range(__size):
20         a.append([0 for i in range(__size)])
21     for i in range(__size):
22         for j in range(__size):
23             if (i == j):
24                 a[i][j] = m[i][j] * -1
25             if (i > j):
26                 a[i][j] = (m[i][j] * -1) - 1
27             if (i < j):
28                 a[i][j] = (m[i][j] * -1) + 1
29
30 def sumar_matriz(m, a):
31     for i in range(__size):
32         for j in range(__size):
33             m[i][j] += a[i][j]
34
35 if __name__ == "__main__":
36     matriz=[]
37     auxiliar=[]
38     llenado_aleatorio(matriz)
39     print("matriz original")
40     mostrar_matriz(matriz)
41
42     generar_auxiliar(matriz, auxiliar)
43     print("matriz generada")
44     mostrar_matriz(auxiliar)
45
46     sumar_matriz(matriz, auxiliar)
47     print("matriz resultante")
48     mostrar_matriz(matriz)
```

Programa 5.24: Lenguaje Python \Leftarrow Ejercicio 89

Ejercicio 90. Escribir un programa que inicialice una matriz de $M(5, 5)$ con números aleatorios entre 1 y 999 y que genere otra matriz $Z(5, 5)$ que contenga en cada celda el conteo de los dígitos de cada número en la matriz M .

```
1 #include <iostream>
2 #include <cstdlib>
3 #include <ctime>
4 using namespace std;
5
6 //procedimiento que llena aleatoriamente la matriz
7 void llenado_aleatorio(int [5][5]);
8
9 //procedimiento que muestra en pantalla la matriz
10 void ver_matriz(int [5][5]);
11
12 //procedimiento que cuenta los dígitos de un entero
13 int contador_recursivo(int);
14
15 //procedimiento que cuenta los dígitos de los elementos de una matriz
16 void contar_digitos_matriz(int [5][5], int [5][5]);
17
18 int main(int argc, char *argv[]) {
19     int matriz[5][5];
20     int auxiliar[5][5];
21
22     srand(time(0));
23     llenado_aleatorio(matriz);
24
25     contar_digitos_matriz(matriz, auxiliar);
26
27     cout << "matriz original" << endl;
28     ver_matriz(matriz);
29
30     cout << "matriz conteo" << endl;
31     ver_matriz(auxiliar);
32
33
34     return 0;
35 }
36
37 void llenado_aleatorio(int m[5][5]){
38     int aleatorio;
39     for(int f=0; f<5; f++){
40         for(int c=0; c<5; c++){
41             aleatorio = 1 + rand() % 998;
42             m[f][c] = aleatorio;
43         }
44         cout << endl;
45     }
46 }
47 void ver_matriz(int m[5][5]){
```

```
48 for(int f=0; f<5; f++){
49     for(int c=0; c<5; c++){
50         if(m[f][c] < 10)
51             cout << " " << m[f][c] << "\t";
52         else if(m[f][c] >= 10 && m[f][c] < 100)
53             cout << " " << m[f][c] << "\t";
54         else
55             cout << m[f][c] << "\t";
56     }
57     cout << endl;
58 }
59 }
60 int contador_recursivo(int numero){
61     if(numero == 0)
62         return 0;
63     else
64         return 1 + contador_recursivo(numero/10);
65 }
66 void contar_digitos_matriz(int m[5][5],int a[5][5]){
67     for(int f=0; f<5; f++){
68         for(int c=0; c<5; c++){
69             a[f][c] = contador_recursivo(m[f][c]);
70         }
71     }
72 }
```

Programa 5.25: Lenguaje C \Leftarrow Ejercicio 90

```
1 from random import randint
2
3 __size = 5
4
5 def llenado_aleatorio(m):
6     for i in range(__size):
7         m.append([randint(1, 999) for i in range(__size)])
8
9 def mostrar_matriz(m):
10    for i in m:
11        for j in i:
12            if j < 10:
13                print(" " + str(j), end='\t')
14            elif j < 100:
15                print(" " + str(j), end='\t')
16            else:
17                print(str(j), end='\t')
18        print("")
19
20 def contador_recursivo(numero):
21     if(numero == 0):
22         return 0
23     else:
24         return 1 + contador_recursivo(int(numero/10))
25
26 def contar_digitos_matriz(m, a):
27     for i in range(__size):
28         a.append([0 for i in range(__size)])
29     for i in range(__size):
30         for j in range(__size):
31             a[i][j] = contador_recursivo(m[i][j])
```

```
32
33 if __name__ == "__main__":
34     matriz=[]
35     auxiliar=[]
36     llenado_aleatorio(matriz)
37     print("matriz original")
38     mostrar_matriz(matriz)
39
40     contar_digitos_matriz(matriz, auxiliar)
41     print("matriz generada")
42     mostrar_matriz(auxiliar)
```

Programa 5.26: Lenguaje Python \Leftarrow Ejercicio 90

Capítulo 6

Manejo de Archivos

Ejercicio 91. El Junquillo Chino

El siguiente problema esta basado en el problema del junquillo chino hallado en el capítulo IX del libro chino: Çhu Chang Suan Shu.^o "Arte Matemático en Nueve Secciones".

Un grajero chino se encontraba mirando un lago circular del cual salía exactamente en el centro una porción de un junquillo, este junquillo nacía exactamente en el centro del fondo del lago, el granjero observó que cuando el viento soplabla el junquillo se inclinaba sin deformarse y al tocar la orilla del lago quedaba exactamente cubierto por el agua. Escribe un programa que le ayude al granjero a determinar la profundidad del lago.

Entrada

La enterada consiste en múltiples líneas de un archivo en la que cada un par de enteros separados por un espacio con $0 < i < 10000$ y $0 < j < 100$ donde i indica el diámetro del lago en metros y j determina la porción visible del junquillo en centímetros. La entrada termina con una línea con un par de ceros.

Salida

Para cada línea de entrada, el programa debe generar la salida redondeada a un decimal de la profundidad del lago en metros.

Ejemplo de entrada:

3 30
10 20
5 25
0 0

Ejemplo de salida:

3,6
62,4
12,4

```
1 def main():
2     entrada = open('ejercicio091.in', 'r')
3
4     while True:
5         linea = entrada.readline()
6         lista = linea.split(" ")
7         if int(lista[0]) == 0 and int(lista[1]) == 0:
8             break
9         else:
10            d, j = (int(lista[0]) * 50, int(lista[1]))
11            x = (d ** 2 - j ** 2) / (2 * j)
12            c = ('{0:.1f}'.format(x / 100))
13            c = c.replace(".", ",")
14            print(c)
15    return 0
16
17    entrada.close()
18
19 if __name__ == '__main__':
20    main()
```

Programa 6.1: Lenguaje Python \Leftarrow Ejercicio 91

Ejercicio 92. Escribir un programa que lea de un archivo una lista de palabras y cuente cuantas veces aparece cada palabra en el texto

```
1 def main():
2     entrada = open('ejercicio092.in', 'r')
3     lista = []
4     palabras = []
5
6     while True:
7         linea = entrada.readline()
8         lista = lista + linea.split(" ")
9         if not linea:
10            break
11    #print(lista)
12    for i in lista:
```

```

13     if not i in palabras and i != '' and i != '\n':
14         palabras.append(i)
15
16     for i in palabras:
17         if lista.count(i) > 1:
18             terminal = " veces"
19         else:
20             terminal = " vez"
21         print("la palabra '" + i + "' aparece " + str(lista.count(i)) +
22               terminal)
23
24     #print(palabras)
25     return 0
26
27     entrada.close()
28
29 if __name__ == '__main__':
30     main()

```

Programa 6.2: Lenguaje Python \Leftarrow Ejercicio 92

Ejercicio 93. Escribir un programa que lea de un archivo una matriz de números enteros de dimensiones $n \times n$ y genere un archivo que contenga la matriz transpuesta de la matriz original

```

1 def main():
2     entrada = open('ejercicio093.in', 'r')
3     salida = open('ejercicio093.out', 'w')
4     lista = []
5     matriz = []
6     traspuesta = []
7
8     while True:
9         linea = entrada.readline()
10        if not linea:
11            break
12        else:
13            lista = linea[:-1].split(" ")
14            matriz.append(lista)
15
16    for f in range(len(matriz)):
17        for c in range(len(matriz[f])):
18            salida.write(matriz[c][f])
19            salida.write(' ')
20        salida.write('\n')
21
22
23    entrada.close()
24    salida.close()
25
26 if __name__ == '__main__':
27     main()

```

Programa 6.3: Lenguaje Python \Leftarrow Ejercicio 93

Ejercicio 94. Escribir un archivo que capture por entrada estándar una serie de números enteros que debe finalizar con un cero y escriba un archivo en donde genere por cada número capturado una línea en la que escriba el número y las veces que fue capturado separados por un espacio en blanco

```
1 def buscar(lista, elemento):
2     x = -1
3     for i in range(len(lista)):
4         if elemento == lista[i][0]:
5             x = i
6     return x
7
8 def main():
9     numeros = []
10
11     while True:
12         numero = int(input("ingrese un numero entero diferente de cero
13 : "))
14         if numero == 0:
15             break
16         else:
17             verificar = buscar(numeros, numero)
18             if verificar == -1:
19                 lista = [numero, 1]
20                 numeros.append(lista)
21             else:
22                 numeros[verificar][1] = numeros[verificar][1] + 1
23
24     salida = open('ejercicio094.out', 'w')
25
26     for f in range(len(numeros)):
27         salida.write(str(numeros[f][0]))
28         salida.write(' -> ')
29         salida.write(str(numeros[f][1]))
30         salida.write('\n')
31     salida.close()
32
33 if __name__ == '__main__':
34     main()
```

Programa 6.4: Lenguaje Python \Leftarrow Ejercicio 94

Ejercicio 95. Una pareja de recién casados compraron un carro y no pueden decidir a donde irán de paseo en estas vacaciones, para decirlo han escrito todos los nombres de los posibles destinos a visitar en un archivo, ayude a la pareja a decidir que destino visitar generando un archivo de tal manera que se asocie a cada lugar un valor entero aleatorio entre 1 y 100 de tal manera que el que tenga el mayor número asociado será el destino a elegir por la pareja

```
1 import random
```

```

2
3 def main():
4     lista = []
5
6     entrada = open('ejercicio095.in', 'r')
7     while True:
8         destino = entrada.readline()
9         destino = destino[:-1]
10        if not destino:
11            break
12        else:
13            lista.append([destino, 0])
14    entrada.close()
15
16    for l in range(len(lista)):
17        peso = random.choice(range(100))
18        peso = peso + 1
19        lista[l][1] = peso
20
21    salida = open('ejercicio095.out', 'w')
22    for f in range(len(lista)):
23        salida.write(str(lista[f][0]))
24        salida.write(' -> ')
25        salida.write(str(lista[f][1]))
26        salida.write('\n')
27    salida.close()
28
29 if __name__ == '__main__':
30     main()

```

Programa 6.5: Lenguaje Python \Leftarrow Ejercicio 95

Ejercicio 96. En un archivo de texto se encuentran emparejadas un serie de palabras de tal manera que la primera consiste en una palabra secreta que un usuario trataba de adivinar y la segunda una serie de letras que el usuario emitio tratando de adivinar la palabra secreta, escribe un programa que le indique al usuario el número de no coincidencias entre las dos palabras

Entrada

La entrada consiste en múltiples líneas en la que cada una contiene un par de cadenas de caracteres separados por un espacio en la que la primera palabra es la palabra secreta y la segunda la palabra del usuario.

Salida

Para cada línea de entrada, el programa debe generar la salida en la que muestre al usuario las dos palabras y en frente y separado por

un espacio el número de no coincidencias de las dos palabras.

Ejemplo de entrada:

casa kza
ferrocarril ferocail
misa lisa

Ejemplo de salida:

casa kza 2
ferrocarril ferocail 0
misa lisa 1

```
1 def comparar(secretas, palabra):
2     contador = 0
3     for c in palabra:
4         if c not in secretas:
5             contador = contador + 1
6     return contador
7
8 def main():
9     entrada = open('ejercicio096.in', 'r')
10    lista = []
11    matriz = []
12
13    while True:
14        linea = entrada.readline()
15        if not linea:
16            break
17        else:
18            lista = linea[:-1].split(" ")
19            matriz.append(lista)
20
21    for e in matriz:
22        print(e[0], e[1], comparar(e[0], e[1]), sep=' ')
23
24    entrada.close()
25
26 if __name__ == '__main__':
27     main()
```

Programa 6.6: Lenguaje Python \Leftarrow Ejercicio 96

Ejercicio 97. Un archivo de texto que se debe enviar por correo electrónico contiene información sensible que no conviene que sea leída por alguien diferente a su destinatario, para garantizar esto, el que envió el archivo codificó este corriendo cada caracter a 4 posiciones hacia adelante en el alfabeto, ayude al lector escribiendo un programa que regrese cada caracter 4 posiciones hacia atrás en el alfabeto de

manera que lo pueda leer, el alfabeto no contiene caracteres especiales como "ñ." "ll" ² todo está escrito en minúscula.

Ejemplo de entrada:

ir pe gewe hi nyere iw pe jmiwx

Ejemplo de salida:

en la casa de juana es la fiesta

```
1 def comparar(secret, palabra):
2     contador = 0
3     for c in palabra:
4         if c not in secret:
5             contador = contador + 1
6     return contador
7
8 def main():
9     entrada = open('ejercicio096.in', 'r')
10    lista = []
11    matriz = []
12
13    while True:
14        linea = entrada.readline()
15        if not linea:
16            break
17        else:
18            lista = linea[:-1].split(" ")
19            matriz.append(lista)
20
21    for e in matriz:
22        print(e[0], e[1], comparar(e[0], e[1]), sep=' ')
23
24    entrada.close()
25
26 if __name__ == '__main__':
27     main()
```

Programa 6.7: Lenguaje Python \Leftarrow Ejercicio 97

Ejercicio 98. Escriba un programa que lea desde un archivo la cantidad de soldados que tendrán dos ejércitos para cada una de las batallas que librarán, el programa debe decir quien tiene mayor probabilidad de ganar cada batalla en base a la cantidad de soldados que tienen, el procesamiento se detiene cuando uno de los dos archivos tiene 0 soldados para lo cual el programa debe generar como salida la palabra "rendicion"

Ejemplo de entrada:

12 2
5 20

10 0

Ejemplo de salida:

gana el 12

gana el 20

rendicion

Ejercicio 99. Escribir un programa que lea desde un archivo una serie de resultados de partidos de beisbol en los que se enfrentaron varios equipos en un torneo todos contra todos, cada entrada consiste en una línea que contiene cuantas carreras anotó el local luego sepados por un espacio el nombre del equipo local luego separado por un espacio el nombre del visitante y por último y separado por un espacio las carreras que anotó el visitante y en base a estos resultados determine el ganador del torneo

Ejemplo de entrada:

cardenales 10 5 magallanes

cardenales 8 7 tigres

cardenales 10 4 leones

leones 4 2 tigres

leones 2 1 magallanes

leones 1 2 cardenales

tigres 4 2 leones

tigres 3 2 magallanes

tigres 1 2 cardenales

magallanes 5 2 tigres

magallanes 3 2 leones

magallanes 6 8 cardenales

Ejemplo de salida:

gana cardenales

Ejercicio 100. Un número feo es todo entero positivo cuyos únicos factores primos son 2, 3 y 5, es decir se pueden escribir como $2^a 3^b 5^c$ con a, b, c enteros positivos o nulos. Los primeros números feos son:

1, 2, 3, 4, 5, 6, 8, 9, 10, 12, 15, 16, 18, 24

Escribir un programa que genere un archivo con los números feos existentes en un rango que se capture desde la entrada estándar.

Bibliografía