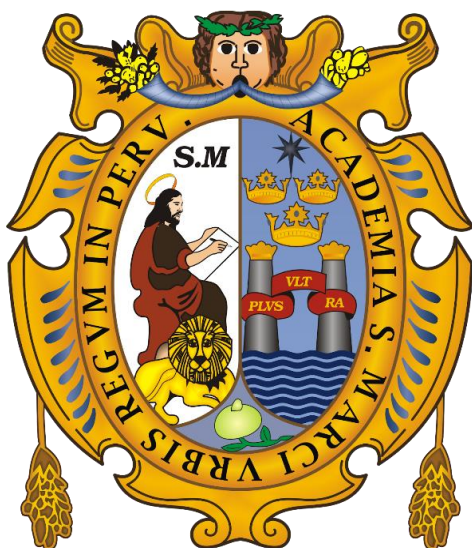


UNIVERSIDAD NACIONAL MAYOR DE SAN MARCOS

Universidad del Perú, la Decana de América



‘INFORME DE EJERCICIOS DE PROGRAMACIÓN’

Facultad de Ciencias Matemáticas

Curso: Algoritmia y Fundamentos de la programación

Profesor: Oscar Benito Pacheco

Integrantes:

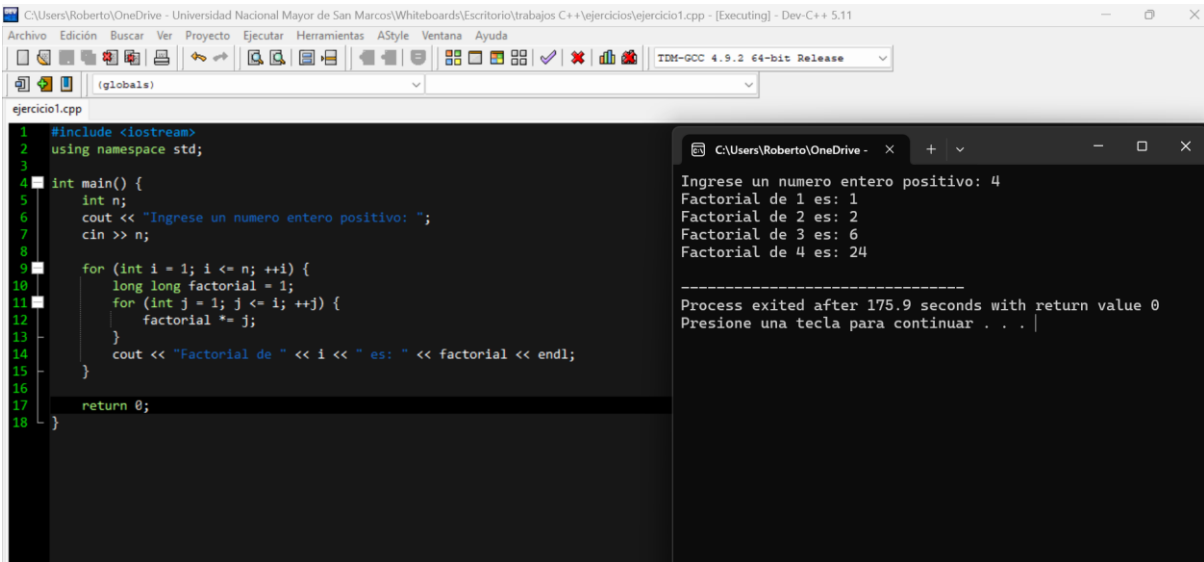
- Palacios Carrasco Roberto (24140173)
- Avalos Palomino Ismael Jesús (24140158)

EJERCICIOS PROPUESTOS (Dev- C++)

Elaborado por: Palacios Carrasco Roberto

Ejercicio1

1.- Hacer un programa que calcule el factorial de todos los números hasta un número dado por teclado



The screenshot shows the Dev-C++ IDE with a C++ program named `ejercicio1.cpp` open. The program calculates the factorial of a number entered by the user. The code is as follows:

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

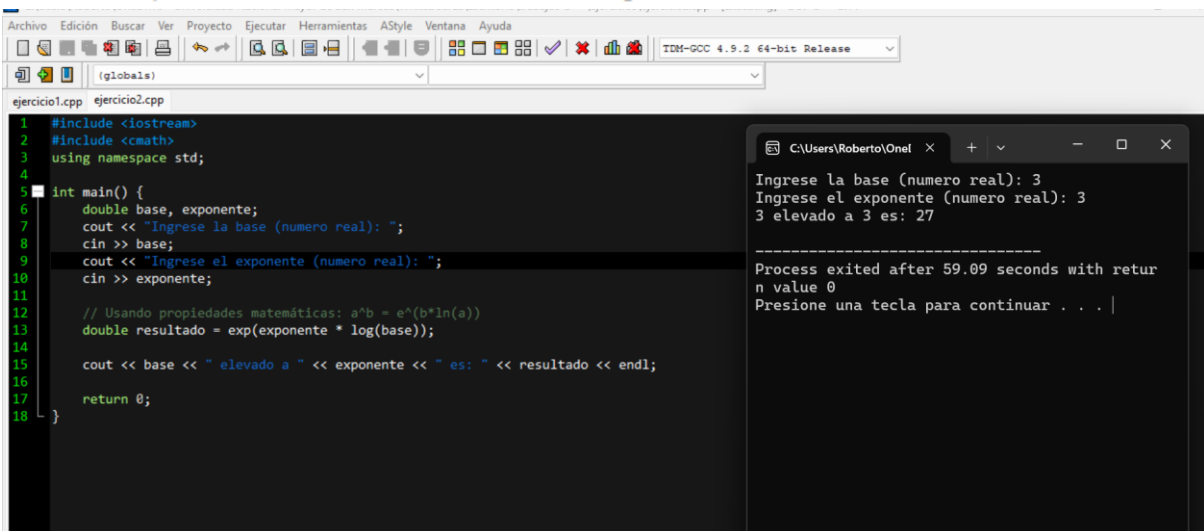
The output window shows the following results for input 4:

```
Ingrese un numero entero positivo: 4
Factorial de 1 es: 1
Factorial de 2 es: 2
Factorial de 3 es: 6
Factorial de 4 es: 24

Process exited after 175.9 seconds with return value 0
Presione una tecla para continuar . . . |
```

Ejercicio2

2.- Realizar un algoritmo que calcule la potencia de un número real elevado a un número real dado su numerador y denominador, sin utilizar funciones ni procedimientos.



The screenshot shows the Dev-C++ IDE with a C++ program named `ejercicio2.cpp` open. The program calculates the power of a real number using the formula $a^b = e^{b \cdot \ln(a)}$. The code is as follows:

```
1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     double base, exponente;
7     cout << "Ingrese la base (numero real): ";
8     cin >> base;
9     cout << "Ingrese el exponente (numero real): ";
10    cin >> exponente;
11
12    // Usando propiedades matemáticas: a^b = e^(b*ln(a))
13    double resultado = exp(exponente * log(base));
14
15    cout << base << " elevado a " << exponente << " es: " << resultado << endl;
16
17    return 0;
18 }
```

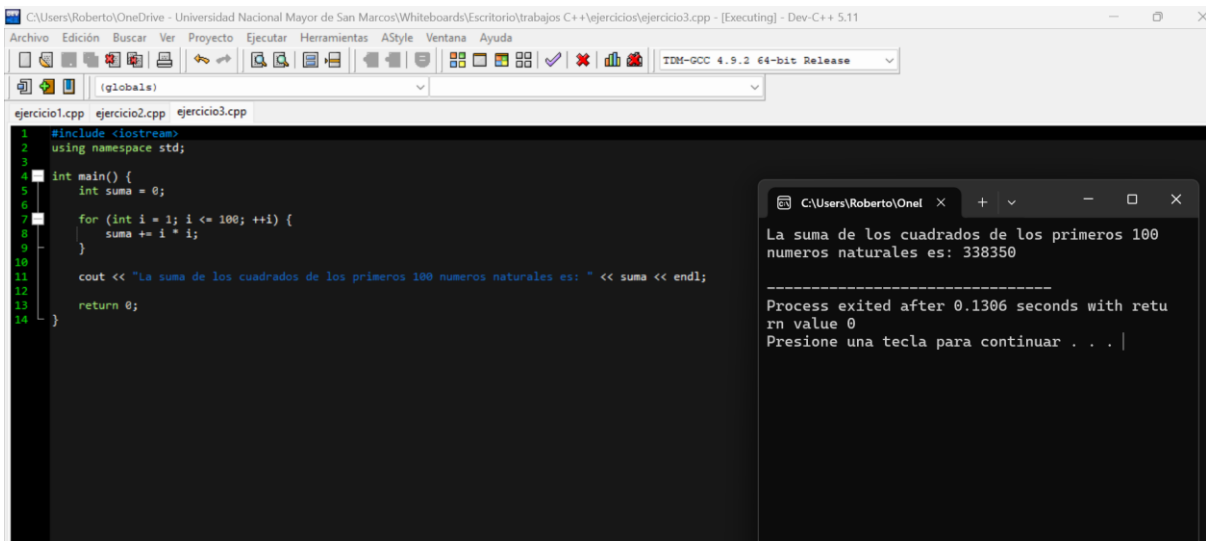
The output window shows the following results for input base 3 and exponent 3:

```
Ingrese la base (numero real): 3
Ingrese el exponente (numero real): 3
3 elevado a 3 es: 27

Process exited after 59.09 seconds with return value 0
Presione una tecla para continuar . . . |
```

Ejercicio3

3.- Construya un programa que imprima la suma de los cuadrados de los 100 primeros números naturales.



The screenshot shows a C++ IDE with a file named `ejercicio3.cpp` open. The code defines a `main` function that initializes a variable `suma` to 0, then uses a `for` loop to calculate the sum of squares of the first 100 natural numbers. The output is displayed in a console window.

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

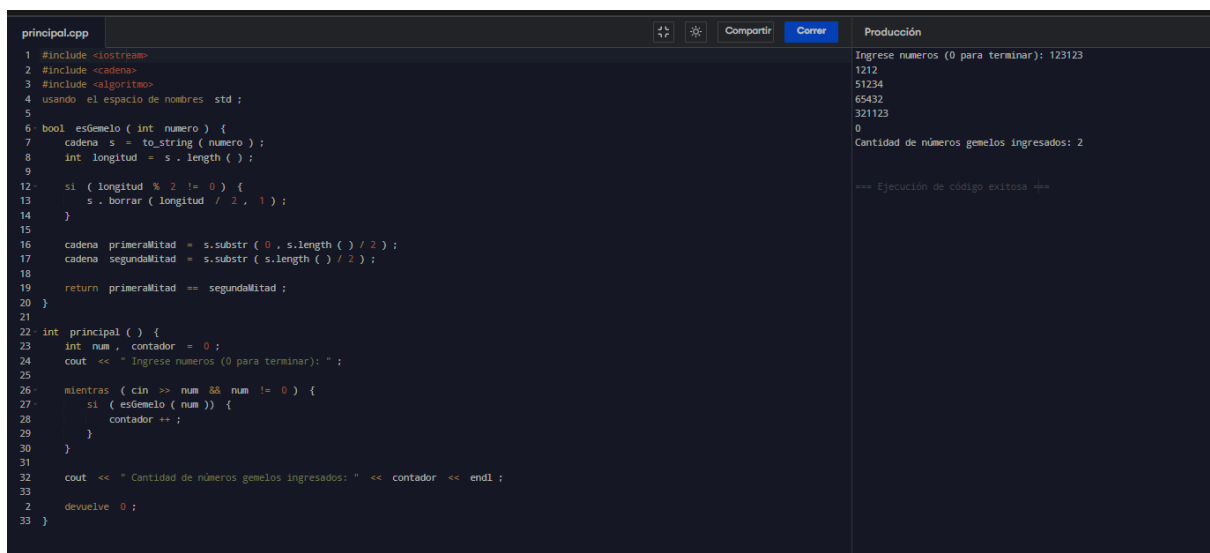
Output:

```
La suma de los cuadrados de los primeros 100
numeros naturales es: 338350

-----
Process exited after 0.1306 seconds with retu
rn value 0
Presione una tecla para continuar . . . |
```

Ejercicio4

4.- Un número gemelo es aquel cuya primera mitad de dígitos es igual a la segunda mitad. Si la cantidad de dígitos es impar, no deberá tomarse en cuenta el dígito del medio Ej. 123123. Hacer un programa para contar los números gemelos de una secuencia de números de entrada.



The screenshot shows a C++ IDE with a file named `principal.cpp` open. The code defines a function `esGemelo` that checks if a number is a palindrome (gemelo) by comparing its first and second halves. The `main` function reads a sequence of numbers from the user and counts how many are palindromic.

```
1 #include <iostream>
2 #include <cadena>
3 #include <algoritmo>
4 usando el espacio de nombres std;
5
6 bool esGemelo ( int numero ) {
7     cadena s = to_string ( numero );
8     int longitud = s . length ( );
9
10
11     si ( longitud % 2 != 0 ) {
12         s . borrar ( longitud / 2 , 1 );
13     }
14
15     cadena primeraMitad = s.substr ( 0 , s.length ( ) / 2 );
16     cadena segundaMitad = s.substr ( s.length ( ) / 2 );
17
18     return primeraMitad == segundaMitad ;
19 }
20
21
22 int principal ( ) {
23     int num , contador = 0 ;
24     cout << " Ingrese numeros (0 para terminar): " ;
25
26     mientras ( cin >> num && num != 0 ) {
27         si ( esGemelo ( num ) ) {
28             contador ++ ;
29         }
30     }
31
32     cout << " Cantidad de números gemelos ingresados: " << contador << endl ;
33
34     devuelve 0 ;
35 }
```

Output:

```
Ingrese numeros (0 para terminar): 123123
1212
51234
65432
321123
0
Cantidad de números gemelos ingresados: 2

=== Ejecución de código exitosa ===
```

Ejercicio5

5.- Calcule la suma y producto de fracciones, antes de realizar las operaciones es necesario que las dos fracciones estén simplificadas a su mínima expresión.

```
ejercicio5.cpp
1 #include <iostream>
2 using namespace std;
3
4 int mcd(int a, int b) {
5     while (b != 0) {
6         int temp = b;
7         b = a % b;
8         a = temp;
9     }
10    return a;
11 }
12
13 void simplificar(int &num, int &den) {
14     int divisor = mcd(num, den);
15     num /= divisor;
16     den /= divisor;
17 }
18
19 int main() {
20     int num1, den1, num2, den2;
21
22     cout << "Ingrese primera fraccion (numerador denominador): ";
23     cin >> num1 >> den1;
24     cout << "Ingrese segunda fraccion (numerador denominador): ";
25     cin >> num2 >> den2;
26
27     // Simplificar fracciones
28     simplificar(num1, den1);
29     simplificar(num2, den2);
30
31     // Suma
32     int sumaNum = num1 * den2 + num2 * den1;
33     int sumaDen = den1 * den2;
34     simplificar(sumaNum, sumaDen);
35
36     // Producto
37     int prodNum = num1 * num2;
38     int prodDen = den1 * den2;
39     simplificar(prodNum, prodDen);
40
41     cout << "Suma: " << sumaNum << "/" << sumaDen << endl;
42     cout << "Producto: " << prodNum << "/" << prodDen << endl;
43     return 0;
44 }
```

```
C:\Users\Roberto\OneDrive - x + v - □ x
Ingrese primera fraccion (numerador denominador): 3
4
Ingrese segunda fraccion (numerador denominador): 1
2
Suma: 5/4
Producto: 3/8

-----
Process exited after 26.29 seconds with return value 0
Presione una tecla para continuar . . .
```

Ejercicio6

6.- Escriba un programa que lea dos fechas dadas por un día, mes y año e indique cuál de ellas es anterior en el tiempo a la otra. El programa seguirá leyendo pares de fechas hasta que se introduzca un valor 0 como día de la primera fecha (en tal caso, **no** seguirá leyendo dicha fecha).

```
ejercicio5.cpp ejercicio6.cpp
1 #include <iostream>
2 using namespace std;
3
4 struct Fecha {
5     int dia;
6     int mes;
7     int ano;
8 };
9
10 bool esAnterior(const Fecha &f1, const Fecha &f2) {
11     if (f1.ano != f2.ano) return f1.ano < f2.ano;
12     if (f1.mes != f2.mes) return f1.mes < f2.mes;
13     return f1.dia < f2.dia;
14 }
15
16 int main() {
17     Fecha fecha1, fecha2;
18
19     while (true) {
20         cout << "Ingrese primera fecha (dia mes año, 0 para terminar): ";
21         cin >> fecha1.dia;
22         if (fecha1.dia == 0) break;
23         cin >> fecha1.mes >> fecha1.ano;
24
25         cout << "Ingrese segunda fecha (dia mes año): ";
26         cin >> fecha2.dia >> fecha2.mes >> fecha2.ano;
27
28         if (esAnterior(fecha1, fecha2)) {
29             cout << "La primera fecha es anterior a la segunda." << endl;
30         } else if (esAnterior(fecha2, fecha1)) {
31             cout << "La segunda fecha es anterior a la primera." << endl;
32         } else {
33             cout << "Las fechas son iguales." << endl;
34         }
35     }
36
37     return 0;
38 }
```

```
C:\Users\Roberto\OneDrive - x + v - □ x
Ingrese primera fecha (dia mes año, 0 para terminar):
25
11
2000
Ingrese segunda fecha (dia mes año): 19
6
1950
La segunda fecha es anterior a la primera.
Ingrese primera fecha (dia mes año, 0 para terminar):
11
9
2008
Ingrese segunda fecha (dia mes año): 10
9
2008
La segunda fecha es anterior a la primera.
Ingrese primera fecha (dia mes año, 0 para terminar):
|
```

Ejercicio7

7.- Modifique el programa anterior para que permita un número máximo de intentos (dado por la constante *MAXINTENTOS*, superados los cuales terminará imprimiendo el mensaje "Lo siento, ha perdido".

```
ejercicio5.cpp  ejercicio6.cpp  ejercicio7.cpp

#include <iostream>
using namespace std;

struct Fecha {
    int dia;
    int mes;
    int ano;
};

bool esAnterior(const Fecha &f1, const Fecha &f2) {
    if (f1.ano != f2.ano) return f1.ano < f2.ano;
    if (f1.mes != f2.mes) return f1.mes < f2.mes;
    return f1.dia < f2.dia;
}

int main() {
    const int MAXINTENTOS = 3;
    int intentos = 0;
    Fecha fecha1, fecha2;

    while (intentos < MAXINTENTOS) {
        cout << "Intentos: " << intentos + 1 << " de " << MAXINTENTOS << endl;
        cout << "Ingrese primera fecha (dia mes aao, 0 para terminar): ";
        cin >> fecha1.dia;
        if (fecha1.dia == 0) break;
        cin >> fecha1.mes >> fecha1.ano;

        cout << "Ingrese segunda fecha (dia mes aao): ";
        cin >> fecha2.dia >> fecha2.mes >> fecha2.ano;

        if (esAnterior(fecha1, fecha2)) {
            cout << "La primera fecha es anterior a la segunda." << endl;
        } else if (esAnterior(fecha2, fecha1)) {
            cout << "La segunda fecha es anterior a la primera." << endl;
        } else {
            cout << "Las fechas son iguales." << endl;
        }

        intentos++;
    }

    if (intentos == MAXINTENTOS) {
        cout << "Lo siento, ha perdido." << endl;
    }

    return 0;
}
```

```
Intento 1 de 3
Ingrese primera fecha (dia mes aao, 0 para terminar)
: 25
11
2000
Ingrese segunda fecha (dia mes aao): 26
11
2000
La primera fecha es anterior a la segunda.
Intento 2 de 3
Ingrese primera fecha (dia mes aao, 0 para terminar)
:11
2
1888
Ingrese segunda fecha (dia mes aao): 12
12
1212
La segunda fecha es anterior a la primera.
Intento 3 de 3
Ingrese primera fecha (dia mes aao, 0 para terminar)
: 10
10
1900
Ingrese segunda fecha (dia mes aao): 3
3
3000
La primera fecha es anterior a la segunda.
Lo siento, ha perdido.
```

Ejercicio9

9.- Modifique el programa anterior para que imprima la tabla de multiplicar completa del 1 al 10.

```
ejercicio5.cpp  ejercicio6.cpp  ejercicio7.cpp  ejercicio9.cpp

#include <iostream>
#include <iomanip>
using namespace std;

int main() {
    cout << "Tabla de multiplicar del 1 al 10:" << endl;
    cout << " ";
    for (int i = 1; i <= 10; ++i) {
        cout << setw(4) << i;
    }
    cout << "\n -----" << endl;

    for (int i = 1; i <= 10; ++i) {
        cout << setw(2) << i << "|";
        for (int j = 1; j <= 10; ++j) {
            cout << setw(4) << i * j;
        }
        cout << endl;
    }

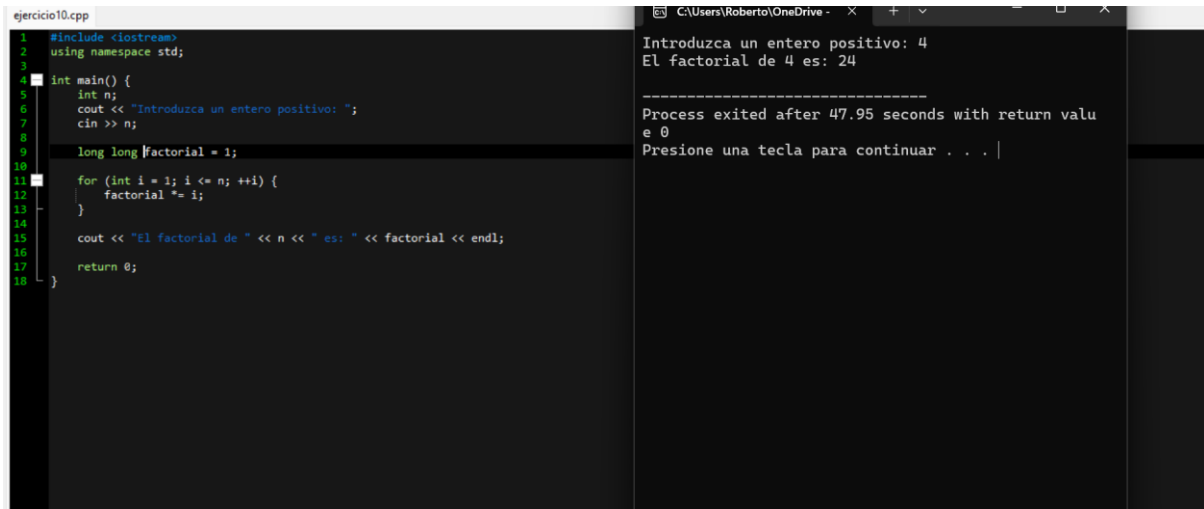
    return 0;
}
```

```
C:\Users\Roberto\OneDrive - x + v - □ X
Tabla de multiplicar del 1 al 10:
      1  2  3  4  5  6  7  8  9 10
-----
1|  1  2  3  4  5  6  7  8  9 10
2|  2  4  6  8 10 12 14 16 18 20
3|  3  6  9 12 15 18 21 24 27 30
4|  4  8 12 16 20 24 28 32 36 40
5|  5 10 15 20 25 30 35 40 45 50
6|  6 12 18 24 30 36 42 48 54 60
7|  7 14 21 28 35 42 49 56 63 70
8|  8 16 24 32 40 48 56 64 72 80
9|  9 18 27 36 45 54 63 72 81 90
10| 10 20 30 40 50 60 70 80 90 100

-----
Process exited after 0.1363 seconds with return value 0
Presione una tecla para continuar . . . |
```

Ejercicio10

10.- El siguiente programa calcula el factorial de un número:



The screenshot shows a C++ program in a code editor and its execution in a terminal. The code defines a function to calculate the factorial of a number using a for loop. The user inputs 4, and the program outputs the factorial of 4, which is 24.

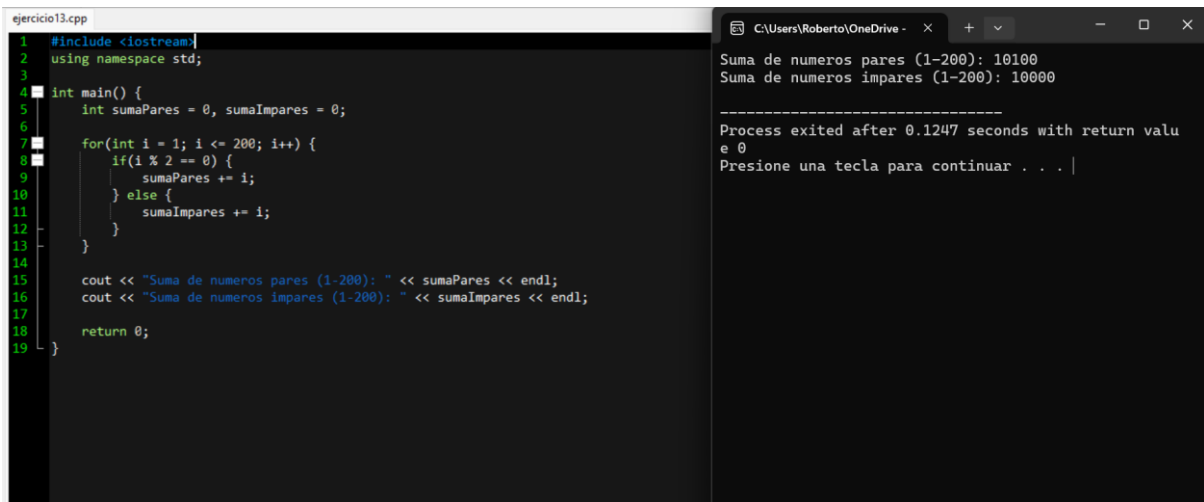
```
ejercicio10.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n;
6     cout << "Introduzca un entero positivo: ";
7     cin >> n;
8
9     long long factorial = 1;
10
11     for (int i = 1; i <= n; ++i) {
12         factorial *= i;
13     }
14
15     cout << "El factorial de " << n << " es: " << factorial << endl;
16
17     return 0;
18 }
```

```
C:\Users\Roberto\OneDrive - x + v
Introduzca un entero positivo: 4
El factorial de 4 es: 24

-----
Process exited after 47.95 seconds with return valu
e 0
Presione una tecla para continuar . . . |
```

Ejercicio13

13.- Se desea calcular independiente la suma de los números pares e impares comprendidos entre 1 y 200.



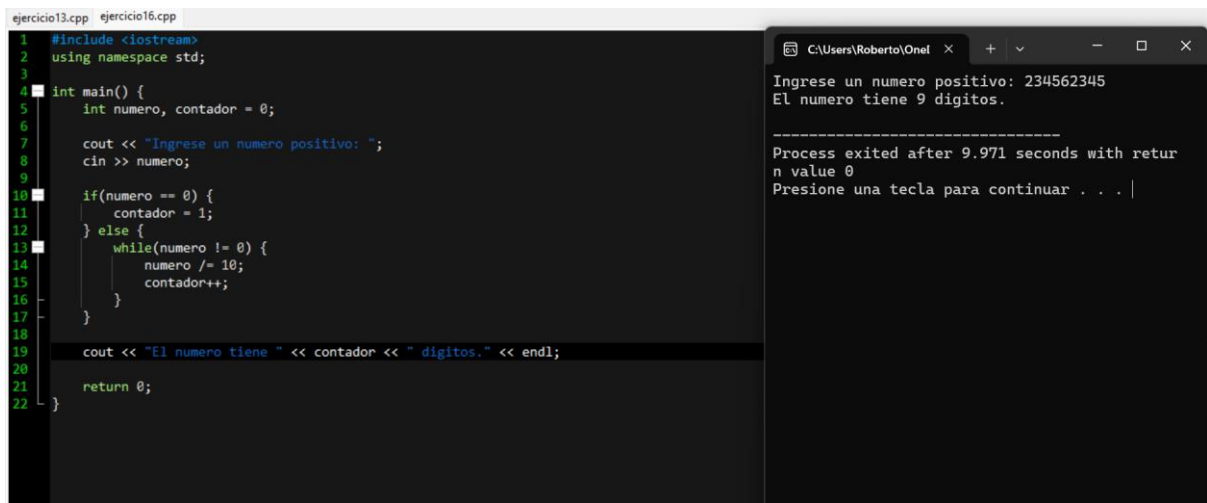
The screenshot shows a C++ program in a code editor and its execution in a terminal. The code uses a for loop to iterate from 1 to 200, checking if each number is even or odd and updating the respective sum. The program outputs the sum of even numbers (10100) and the sum of odd numbers (10000).

```
ejercicio13.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int sumaPares = 0, sumaImpares = 0;
6
7     for (int i = 1; i <= 200; i++) {
8         if (i % 2 == 0) {
9             sumaPares += i;
10         } else {
11             sumaImpares += i;
12         }
13     }
14
15     cout << "Suma de numeros pares (1-200): " << sumaPares << endl;
16     cout << "Suma de numeros impares (1-200): " << sumaImpares << endl;
17
18     return 0;
19 }
```

```
C:\Users\Roberto\OneDrive - x + v
Suma de numeros pares (1-200): 10100
Suma de numeros impares (1-200): 10000

-----
Process exited after 0.1247 seconds with return valu
e 0
Presione una tecla para continuar . . . |
```

Ejercicio16



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int numero, contador = 0;
6
7     cout << "Ingrese un numero positivo: ";
8     cin >> numero;
9
10    if(numero == 0) {
11        contador = 1;
12    } else {
13        while(numero != 0) {
14            numero /= 10;
15            contador++;
16        }
17    }
18
19    cout << "El numero tiene " << contador << " digitos." << endl;
20
21    return 0;
22 }
```

C:\Users\Roberto\OneL x + - □ ×

Ingrese un numero positivo: 234562345
El numero tiene 9 digitos.

Process exited after 9.971 seconds with return value 0
Presione una tecla para continuar . . . |

Ejercicio19

19.- Trazar la ejecución del siguiente algoritmo, y hallar el resultado producido cuando los datos de entrada son: 49 y 70.

ALGORITMO Divisor

VARIABLES

Z primero, segundo

INICIO

Leer (primero, segundo)

MIENTRAS <> segundo HACER

SI primero > segundo ENTONCES

primero = primero - segundo

SINO

segundo = segundo - primero

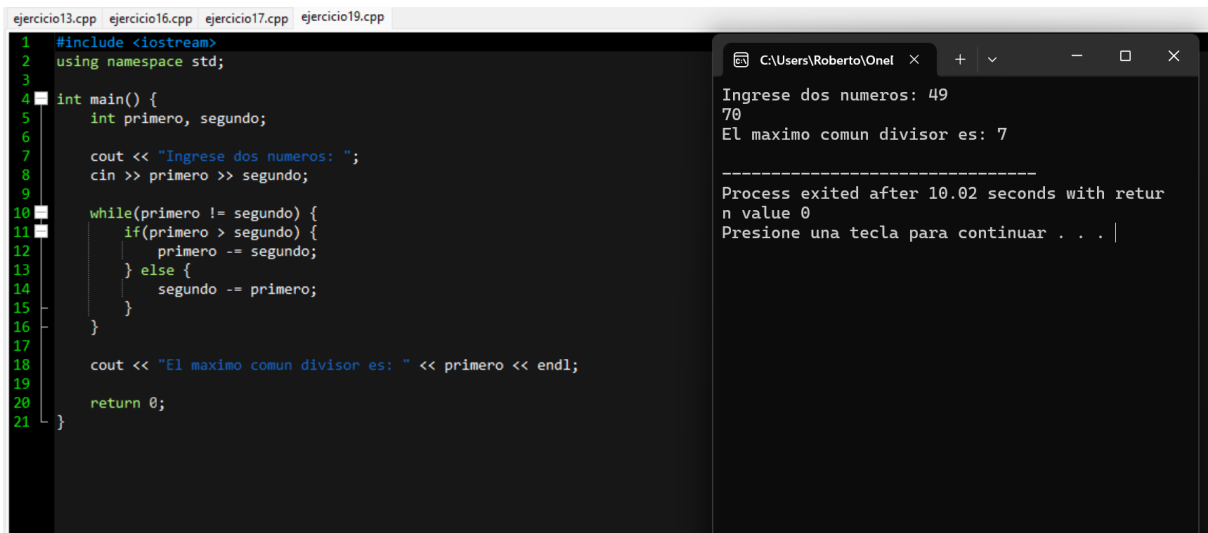
FINSI

FINMIENTRAS

Escribir (primero)

FIN Divisor

Escriba su programa fuente en C++



```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int primero, segundo;
6
7     cout << "Ingrese dos numeros: ";
8     cin >> primero >> segundo;
9
10    while(primero != segundo) {
11        if(primero > segundo) {
12            primero -= segundo;
13        } else {
14            segundo -= primero;
15        }
16    }
17
18    cout << "El maximo comun divisor es: " << primero << endl;
19
20    return 0;
21 }
```

C:\Users\Roberto\OneL x + - □ ×

Ingrese dos numeros: 49
70
El maximo comun divisor es: 7

Process exited after 10.02 seconds with return value 0
Presione una tecla para continuar . . . |

Ejercicio21

21.- Elabore un programa que lea un número entero y escriba el número resultante de invertir sus cifras.

```
ejercicio21.cpp
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int numero, invertido = 0;
6
7     cout << "Ingrese un número entero: ";
8     cin >> numero;
9
10    while(numero != 0) {
11        invertido = invertido * 10 + numero % 10;
12        numero /= 10;
13    }
14
15    cout << "Número invertido: " << invertido << endl;
16
17    return 0;
18 }
```

```
C:\Users\Roberto\OneL x + - □ x
Ingrese un número entero: 21231
Número invertido: 13212

-----
Process exited after 20.29 seconds with return value 0
Presione una tecla para continuar . . . |
```

Ejercicio22

22.- Elabore un programa C++ que lea por teclado un número n entero positivo y presente por pantalla una pirámide de n filas que responda al siguiente esquema:

```
1
2 3 2
3 4 5 4 3
4 5 6 7 6 5 4
5 6 7 8 9 8 7 6 5
6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
9 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 9
0 1 2 3 4 5 6 7 8 9 8 7 6 5 4 3 2 1 0
```

```
ejercicio21.cpp ejercicio22.cpp
1 #include <iostream>
2 #include <iomanip>
3 using namespace std;
4
5 int main() {
6     int n;
7     cout << "Ingrese número de filas | ";
8     cin >> n;
9
10    for(int i = 1; i <= n; i++) {
11        // Espacios iniciales
12        cout << setw(2*(n-i)) << " ";
13
14        // Números crecientes
15        for(int j = i; j <= 2*i-1; j++) {
16            cout << j%10 << " ";
17        }
18
19        // Números decrecientes
20        for(int j = 2*i-2; j >= i; j--) {
21            cout << j%10 << " ";
22        }
23
24        cout << endl;
25    }
26
27    return 0;
28 }
```

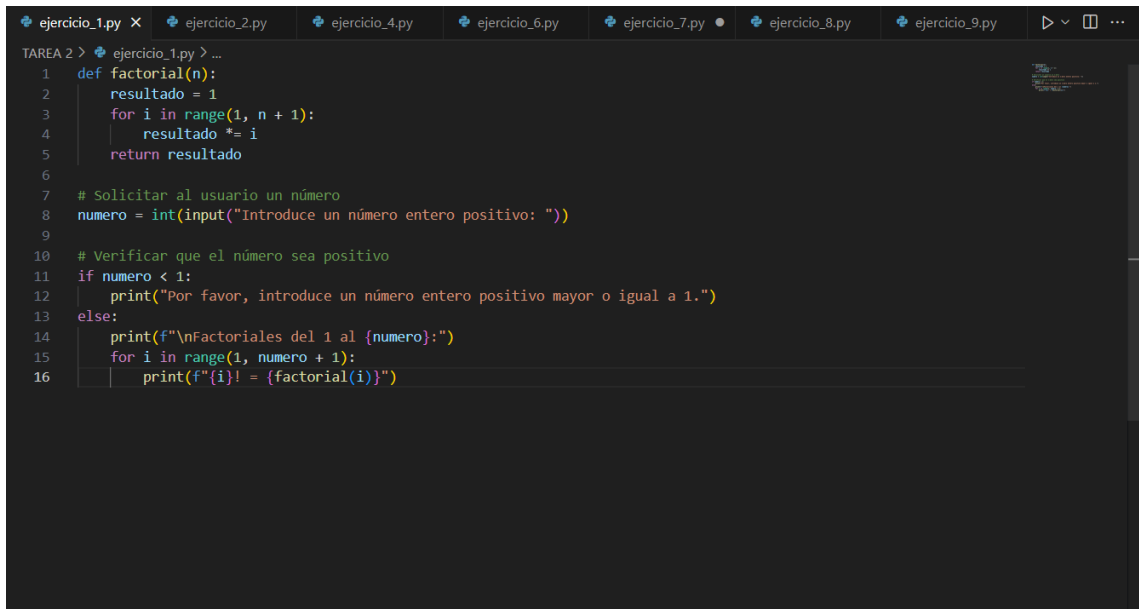
```
C:\Users\Roberto\OneL x + - □ x
Ingrese número de filas : 9
      1
     2 3 2
    3 4 5 4 3
   4 5 6 7 6 5 4
  5 6 7 8 9 8 7 6 5
 6 7 8 9 0 1 0 9 8 7 6
7 8 9 0 1 2 3 2 1 0 9 8 7
8 9 0 1 2 3 4 5 4 3 2 1 0 9 8
9 0 1 2 3 4 5 6 7 6 5 4 3 2 1 0 9

-----
Process exited after 6.494 seconds with return value 0
Presione una tecla para continuar . . . |
```

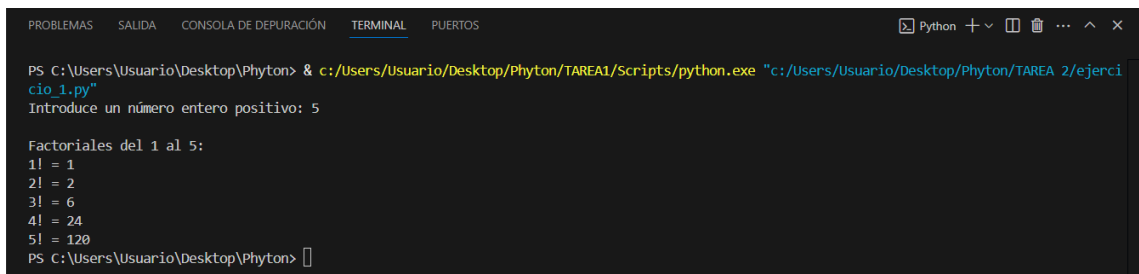

EJERCICIOS PROPUESTOS (Python)

Elaborado por: Avalos Palomino Ismael Jesús

Tarea1:



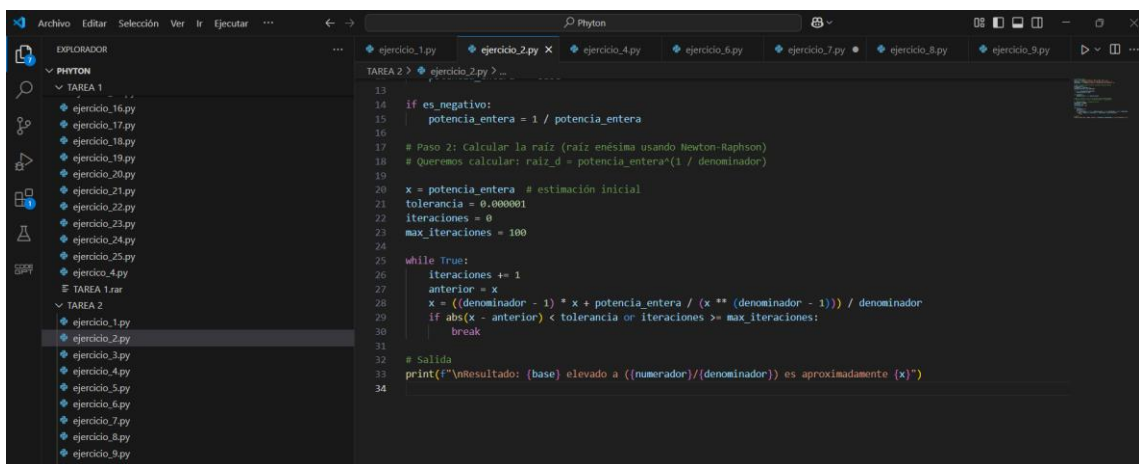
```
TAREA 2 > ejercicio_1.py > ...
1 def factorial(n):
2     resultado = 1
3     for i in range(1, n + 1):
4         resultado *= i
5     return resultado
6
7 # Solicitar al usuario un número
8 numero = int(input("Introduce un número entero positivo: "))
9
10 # Verificar que el número sea positivo
11 if numero < 1:
12     print("Por favor, introduce un número entero positivo mayor o igual a 1.")
13 else:
14     print(f"\nFactoriales del 1 al {numero}:")
15     for i in range(1, numero + 1):
16         print(f"{i}! = {factorial(i)}")
```



```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_1.py"
Introduce un número entero positivo: 5

Factoriales del 1 al 5:
1! = 1
2! = 2
3! = 6
4! = 24
5! = 120
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 2:



```
TAREA 2 > ejercicio_2.py > ...
13
14 if es_negativo:
15     potencia_entera = 1 / potencia_entera
16
17 # Paso 2: Calcular la raíz (raíz enésima usando Newton-Raphson)
18 # Queremos calcular: raíz_n = potencia_entera**(1 / denominador)
19
20 x = potencia_entera # estimación inicial
21 tolerancia = 0.000001
22 iteraciones = 0
23 max_iteraciones = 100
24
25 while True:
26     iteraciones += 1
27     anterior = x
28     x = ((denominador - 1) * x + potencia_entera / (x ** (denominador - 1))) / denominador
29     if abs(x - anterior) < tolerancia or iteraciones >= max_iteraciones:
30         break
31
32 # Salida
33 print(f"\nResultado: (base) elevado a {(numerador)/(denominador)} es aproximadamente (x)")
34
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_2.py"
Introduce la base (número real): 8
Introduce el numerador del exponente: 4
Introduce el denominador del exponente: 9

Resultado: 8.0 elevado a (4/9) es aproximadamente 2.5198420997898845
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 3:

```
ejercicio_5.py  ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py X  ejercicio_6.py  ejercicio_7.py  ▶ ▢ ...

TAREA 2 > ejercicio_3.py > ...
1  def suma_de_cuadrados(n):
2      suma = 0
3      for i in range(1, n + 1):
4          suma += i ** 2
5      return suma
6
7  # Ejemplo de uso
8  n = 100
9  resultado = suma_de_cuadrados(n)
10 print(f"La suma de los cuadrados de los primeros {n} números naturales es: {resultado}")
11
12
13
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_2.py"
Introduce la base (número real): 8
Introduce el numerador del exponente: 4
Introduce el denominador del exponente: 9

Resultado: 8.0 elevado a (4/9) es aproximadamente 2.5198420997898845
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 4:

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_3.py"
La suma de los cuadrados de los primeros 100 números naturales es: 338350
PS C:\Users\Usuario\Desktop\Phyton>
```

```
ejercicio_5.py  ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py X  ejercicio_6.py  ejercicio_7.py  ▶ ▢ ...
TAREA 2 > ejercicio_3.py > ...
1  def suma_de_cuadrados(n):
2      suma = 0
3      for i in range(1, n + 1):
4          suma += i ** 2
5      return suma
6
7  # Ejemplo de uso
8  n = 100
9  resultado = suma_de_cuadrados(n)
10 print(f"La suma de los cuadrados de los primeros {n} números naturales es: {resultado}")
11
12
13
```

Tarea 5:

```
ejercicio_5.py X  ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py  ejercicio_6.py  ejercicio_7.py  ▶ ▢ ...
TAREA 2 > ejercicio_5.py > Fraccion
1  class Fraccion:
2      def __init__(self, numerador, denominador):
3          self.numerador = numerador
4          self.denominador = denominador
5          self.simplificar()
6
7      def simplificar(self):
8          def gcd(a, b):
9              while b:
10                 a, b = b, a % b
11             return a
12
13         comun_divisor = gcd(self.numerador, self.denominador)
14         self.numerador //= comun_divisor
15         self.denominador //= comun_divisor
16
17     def __add__(self, otra):
18         nuevo_numerador = self.numerador * otra.denominador + otra.numerador * self.denominador
19         nuevo_denominador = self.denominador * otra.denominador
20         return Fraccion(nuevo_numerador, nuevo_denominador)
21
22     def __mul__(self, otra):
23         nuevo_numerador = self.numerador * otra.numerador
24         nuevo_denominador = self.denominador * otra.denominador
25         return Fraccion(nuevo_numerador, nuevo_denominador)
26
27     def __str__(self):
28         (variable) denominador: Any
29         return f"{self.numerador}/{self.denominador}"
30
31 # Ejemplo de uso
32 f1 = Fraccion(1, 2)
33 f2 = Fraccion(1, 3)
34
35 suma = f1 + f2
36 producto = f1 * f2
37
38 print(f"Suma: {suma}")
39 print(f"Producto: {producto}")
40
41
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

Python + -  [ ] [ ] ... ^ x

PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejerci
cio_5.py"
Suma: 5/6
Producto: 1/6
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 6:

```
ejercicio_5.py  ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py  ejercicio_6.py x  ejercicio_7.py  [ ] [ ] ...

TAREA 2 > ejercicio_6.py > ...
1  while True:
2      print("\nIntroduce la primera fecha:")
3      dia1 = int(input("Día: "))
4      if dia1 == 0:
5          print("\nFin del programa.")
6          break
7      mes1 = int(input("Mes: "))
8      anio1 = int(input("Año: "))
9
10     print("Introduce la segunda fecha:")
11     dia2 = int(input("Día: "))
12     mes2 = int(input("Mes: "))
13     anio2 = int(input("Año: "))
14
15     # Comparar fechas: primero el año, luego mes, luego día
16     if (anio1, mes1, dia1) < (anio2, mes2, dia2):
17         print("La primera fecha es anterior.")
18     elif (anio1, mes1, dia1) > (anio2, mes2, dia2):
19         print("La segunda fecha es anterior.")
20     else:
21         print("Ambas fechas son iguales.")
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS

Python + -  [ ] [ ] ... ^ x

Introduce la primera fecha:
Día: 5
Mes: 4
Año: 2016
Introduce la segunda fecha:
Día: 15
Mes: 4
Año: 2007
La segunda fecha es anterior.
```

Tarea 7:

```
ejercicio_5.py  ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py  ejercicio_6.py  ejercicio_7.py  ▾ □ ...
TAREA 2 > ejercicio_7.py > ...
1  # Constante con el número máximo de intentos
2  MAXINTENTOS = 3
3  intentos = 0
4
5  while intentos < MAXINTENTOS:
6      print(f"\nIntento {intentos + 1} de {MAXINTENTOS}")
7      print("Introduce la primera fecha:")
8      dia1 = int(input("Día: "))
9      if dia1 == 0:
10         print("\nFin del programa.")
11         break
12     mes1 = int(input("Mes: "))
13     anio1 = int(input("Año: "))
14
15     print("Introduce la segunda fecha:")
16     dia2 = int(input("Día: "))
17     mes2 = int(input("Mes: "))
18     anio2 = int(input("Año: "))
19
20     # Comparar fechas
21     if (anio1, mes1, dia1) < (anio2, mes2, dia2):
22         print("La primera fecha es anterior.")
23     elif (anio1, mes1, dia1) > (anio2, mes2, dia2):
24         print("La segunda fecha es anterior.")
25     else:
26         print("Ambas fechas son iguales.")
27
28     intentos += 1
29
30 if intentos >= MAXINTENTOS:
31     print("\nLo siento, ha perdido.")
32
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS  Python + ▾ □ ... ^ ×
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_7.py"

Intento 1 de 3
Introduce la primera fecha:
Día: 5
Mes: 7
Año: 2015
Introduce la segunda fecha:
Día: 8
Mes: 4
Año: 2014
La segunda fecha es anterior.

Intento 2 de 3
Introduce la primera fecha:
Día: 6
Mes: 12
Año: 2005
Introduce la segunda fecha:
Día: 3
Mes: 7
Año: 2007
La primera fecha es anterior.

Intento 3 de 3
Introduce la primera fecha:
Día: 8
Mes: 6
Año: 2016
Introduce la segunda fecha:
Día: 25
Mes: 4
Año: 1894
La segunda fecha es anterior.

Lo siento, ha perdido.
```

Tarea 8:

```
ejercicio_1.py  ejercicio_2.py  ejercicio_4.py  ejercicio_3.py  ejercicio_6.py  ejercicio_7.py  ejercicio_8.py x
TAREA 2 > ejercicio_8.py > ...
17
18     print("Introduce la segunda fecha:")
19     dia2 = int(input("Día: "))
20     mes2 = int(input("Mes: "))
21     anio2 = int(input("Año: "))
22
23     # Comparar fechas
24     if (anio1, mes1, dia1) < (anio2, mes2, dia2):
25         print("La primera fecha es anterior.")
26     elif (anio1, mes1, dia1) > (anio2, mes2, dia2):
27         print("La segunda fecha es anterior.")
28     else:
29         print("Ambas fechas son iguales.")
30
31     # Aumentar el contador de intentos
32     intentos += 1
33
34     # Si se han agotado los intentos, mostrar mensaje final
35     if intentos >= MAXINTENTOS:
36         print("\nLo siento, ha perdido.")
37
```

```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
Introduce la segunda fecha:
Día: 6
Mes: 8
Año: 2001
La segunda fecha es anterior.

Intento 3 de 3
Introduce la primera fecha:
Día: 9
Mes: 5
Año: 1085
Introduce la segunda fecha:
Día: 5
Mes: 5
Año: 231
La segunda fecha es anterior.

Lo siento, ha perdido.
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 9:

```
TAREA 2 > ejercicio_9.py > ...
1     print("Tabla de multiplicar del 1 al 10:\n")
2
3     for i in range(1, 11): # Filas
4         for j in range(1, 11): # Columnas
5             resultado = i * j
6             print(f"{i} x {j} = {resultado}".ljust(15), end='')
7         print() # Salto de línea por cada fila
8
```

PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN	TERMINAL	PUERTOS	Python + - [icon] ... ^ X					
1 x 1 = 1 10	1 x 2 = 2	1 x 3 = 3	1 x 4 = 4	1 x 5 = 5	1 x 6 = 6	1 x 7 = 7	1 x 8 = 8	1 x 9 = 9	1 x 10 =	
2 x 1 = 2 20	2 x 2 = 4	2 x 3 = 6	2 x 4 = 8	2 x 5 = 10	2 x 6 = 12	2 x 7 = 14	2 x 8 = 16	2 x 9 = 18	2 x 10 =	
3 x 1 = 3 30	3 x 2 = 6	3 x 3 = 9	3 x 4 = 12	3 x 5 = 15	3 x 6 = 18	3 x 7 = 21	3 x 8 = 24	3 x 9 = 27	3 x 10 =	
4 x 1 = 4 40	4 x 2 = 8	4 x 3 = 12	4 x 4 = 16	4 x 5 = 20	4 x 6 = 24	4 x 7 = 28	4 x 8 = 32	4 x 9 = 36	4 x 10 =	
5 x 1 = 5 50	5 x 2 = 10	5 x 3 = 15	5 x 4 = 20	5 x 5 = 25	5 x 6 = 30	5 x 7 = 35	5 x 8 = 40	5 x 9 = 45	5 x 10 =	
6 x 1 = 6 60	6 x 2 = 12	6 x 3 = 18	6 x 4 = 24	6 x 5 = 30	6 x 6 = 36	6 x 7 = 42	6 x 8 = 48	6 x 9 = 54	6 x 10 =	
7 x 1 = 7 70	7 x 2 = 14	7 x 3 = 21	7 x 4 = 28	7 x 5 = 35	7 x 6 = 42	7 x 7 = 49	7 x 8 = 56	7 x 9 = 63	7 x 10 =	
8 x 1 = 8 80	8 x 2 = 16	8 x 3 = 24	8 x 4 = 32	8 x 5 = 40	8 x 6 = 48	8 x 7 = 56	8 x 8 = 64	8 x 9 = 72	8 x 10 =	
9 x 1 = 9 90	9 x 2 = 18	9 x 3 = 27	9 x 4 = 36	9 x 5 = 45	9 x 6 = 54	9 x 7 = 63	9 x 8 = 72	9 x 9 = 81	9 x 10 =	
10 x 1 = 10 = 100	10 x 2 = 20	10 x 3 = 30	10 x 4 = 40	10 x 5 = 50	10 x 6 = 60	10 x 7 = 70	10 x 8 = 80	10 x 9 = 90	10 x 10 =	
PS C:\Users\Usuario\Desktop\Phyton>										

Tarea 10:

```

TAREA 2 > ejercicio_10.py > ...
1 # Versión con estructura "para" (for)
2 n = int(input("Introduzca un entero positivo: "))
3 factorial = 1
4
5 for i in range(1, n + 1):
6     factorial *= i
7
8 print(f"El factorial de {n} es: {factorial}")
9
10

```

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + - [icon] ... ^ X

PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejerci
cio_10.py"
Introduzca un entero positivo: 5
El factorial de 5 es: 120
PS C:\Users\Usuario\Desktop\Phyton>

```

Tarea 11:

```
TAREA 2 > ● ejercicio_11.py > ...
1  import math
2
3  # Definir centros y radios
4  c1_centro = (5, 4)
5  c1_radio = 2
6
7  c2_centro = (-5, -4)
8  c2_radio = 3
9
10 # Lista de puntos de prueba (puedes modificarla o leer desde input)
11 puntos = [
12     (4, 4),      # dentro de c1
13     (5, 6),      # sobre la frontera de c1
14     (-6, -4),    # dentro de c2
15     (-2, -2),    # fuera de ambas
16     (-5, -1),    # sobre la frontera de c2
17     (0, 0)       # fuera de ambas
18 ]
19
20 # Función para verificar si un punto está dentro o sobre una circunferencia
21 def dentro_o_sobre(punto, centro, radio):
22     x, y = punto
23     cx, cy = centro
24     return (x - cx)**2 + (y - cy)**2 <= radio**2
25
26 # Contadores
27 dentro_c1 = 0
28 dentro_c2 = 0
29
30 for punto in puntos:
31     if dentro_o_sobre(punto, c1_centro, c1_radio):
32         dentro_c1 += 1
33     if dentro_o_sobre(punto, c2_centro, c2_radio):
34         dentro_c2 += 1
35
36 print(f"Puntos dentro o sobre la circunferencia c1: {dentro_c1}")
37 print(f"Puntos dentro o sobre la circunferencia c2: {dentro_c2}")
38
```




```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACIÓN  TERMINAL  PUERTOS
Python + - [icon] [icon] [icon] [icon] [icon] [icon]
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_11.py"
Puntos dentro o sobre la circunferencia c1: 2
Puntos dentro o sobre la circunferencia c2: 2
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 12:

```

TAREA 2 > ➤ ejercicio_12.py > ...
1  # Cifrado
2  def cifrar(numero):
3      # Convertir el número en una lista de dígitos
4      digitos = [int(d) for d in str(numero)]
5
6      # Aplicar el cifrado a cada dígito
7      cifrado = [(d + 7) % 10 for d in digitos]
8
9      # Intercambiar el primer y tercer dígito, el segundo y cuarto
10     cifrado[0], cifrado[2] = cifrado[2], cifrado[0]
11     cifrado[1], cifrado[3] = cifrado[3], cifrado[1]
12
13     # Convertir de nuevo a un número entero
14     numero_cifrado = ''.join(map(str, cifrado))
15
16     return numero_cifrado
17
18 # Lectura del número a cifrar
19 numero = int(input("Introduce un número de 4 dígitos: "))
20 numero_cifrado = cifrar(numero)
21 print(f"El número cifrado es: {numero_cifrado}")


```



```
PROBLEMAS  SALIDA  CONSOLA DE DEPURACION  TERMINAL  PUERTOS
PS C:\Users\Usuario\Desktop\Phyton> & c:\Users\Usuario\Desktop\Phyton\TAREA1\Scripts\python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_12.py"
Introduce un número de 4 dígitos: 4564
El número cifrado es: 3112
PS C:\Users\Usuario\Desktop\Phyton>
```


Tarea 13:


```
TAREA 2 > ejercicio_13.py > ...
1 # Inicializar las sumas
2 suma_pares = 0
3 suma_impares = 0
4
5 # Iterar sobre los números del 1 al 200
6 for i in range(1, 201):
7     if i % 2 == 0:
8         suma_pares += i # Si el número es par
9     else:
10        suma_impares += i # Si el número es impar
11
12 # Mostrar los resultados
13 print(f"La suma de los números pares entre 1 y 200 es: {suma_pares}")
14 print(f"La suma de los números impares entre 1 y 200 es: {suma_impares}")
```



```
PS C:\Users\Usuario\Desktop\Phyton> c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_13.py"
La suma de los números pares entre 1 y 200 es: 10100
La suma de los números impares entre 1 y 200 es: 10000
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 14:

```
TAREA 2 > ejercicio_14.py > ...
1 # Inicializar la suma
2 suma_cuadrados = 0
3
4 # Iterar sobre los primeros 100 números naturales
5 for i in range(1, 101):
6     suma_cuadrados += i**2 # Sumar el cuadrado de i
7
8 # Mostrar el resultado
9 print(f"La suma de los cuadrados de los primeros 100 números naturales es: {suma_cuadrados}")
10
```



The screenshot shows a Windows command prompt window with the following text:

```

PS C:\Users\Usuario\Desktop\Phyton> & c:\Users\Usuario\Desktop\Phyton\TAREA1\Scripts\python.exe "c:\Users\Usuario\Desktop\Phyton\TAREA 2\ejercicio 14.py"
La suma de los cuadrados de los primeros 100 números naturales es: 338350
PS C:\Users\Usuario\Desktop\Phyton>

```

Tarea 15:

```

TAREA 2 > ejercicio_15.py > ...
1  # Función para calcular la intersección de dos intervalos
2  def interseccion_intervalos(a, b, c, d):
3      # El punto de inicio de la intersección es el máximo entre a y c
4      inicio = max(a, c)
5      # El punto final de la intersección es el mínimo entre b y d
6      fin = min(b, d)
7
8      # Verificar si la intersección es válida (el inicio debe ser menor o igual al fin)
9      if inicio <= fin:
10         return (inicio, fin) # Devuelve la intersección
11     else:
12         return None # Si no hay intersección, devuelve None
13
14 # Ejemplo de uso
15 a, b = map(int, input("Introduce el primer intervalo [a, b] (separados por espacio): ").split())
16 c, d = map(int, input("Introduce el segundo intervalo [c, d] (separados por espacio): ").split())
17
18 interseccion = interseccion_intervalos(a, b, c, d)
19
20 if interseccion:
21     print(f"La intersección de los intervalos es: {interseccion}")
22 else:
23     print("Los intervalos no se intersectan.")

```

```
PROBLEMAS  SALIDA  TERMINAL  PUERTOS
>  TERMINAL
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_15.py"
Introduce el primer intervalo [a, b] (separados por espacio): 2 4
Introduce el segundo intervalo [c, d] (separados por espacio): 4 5
La intersección de los intervalos es: (4, 4)
PS C:\Users\Usuario\Desktop\Phyton>
```

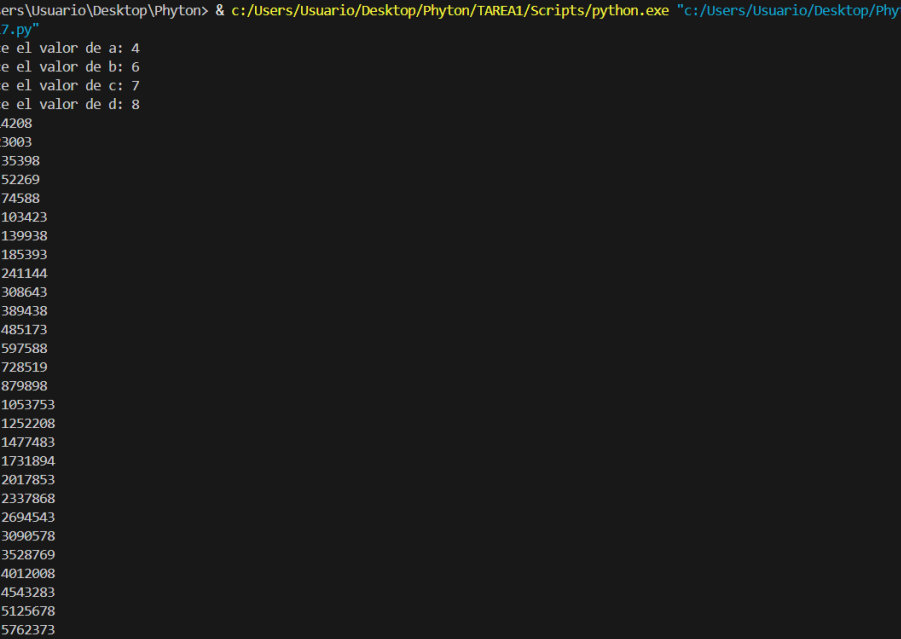
Tarea 16:

```
TAREA 2 > ejercicio_16.py > ...
1  # Leer el número positivo N
2  n = int(input("Introduce un número positivo: "))
3
4  # Determinar el número de dígitos
5  numero_digitos = len(str(n))
6
7  # Mostrar el resultado
8  print(f"El número {n} tiene {numero_digitos} dígitos.")
9
```

```
PROBLEMAS  SALIDA  TERMINAL  PUERTOS
>  TERMINAL
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_16.py"
Introduce un número positivo: 56541
El número 56541 tiene 5 dígitos.
PS C:\Users\Usuario\Desktop\Phyton>
```

Tarea 17:

```
TAREA 2 > ejercicio_17.py > ...
1  # Leer los coeficientes a, b, c, d
2  a = int(input("Introduce el valor de a: "))
3  b = int(input("Introduce el valor de b: "))
4  c = int(input("Introduce el valor de c: "))
5  d = int(input("Introduce el valor de d: "))
6
7  # Evaluar la función para x en el rango [8, 100]
8  for x in range(8, 101):
9      # Calcular f(x) = aX^4 - bX^3 + 2cX^2 - X + d
10     f_x = a * x**4 - b * x**3 + 2 * c * x**2 - x + d
11
12     # Mostrar el resultado para cada x
13     print(f"f({x}) = {f_x}")
14
```



```
PROBLEMAS SOLIDA TERMINAL PUERTOS
> v TERMINAL Python + ~ [] ...
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio 17.py"
Introduce el valor de a: 4
Introduce el valor de b: 6
Introduce el valor de c: 7
Introduce el valor de d: 8
f(8) = 14208
f(9) = 23003
f(10) = 35398
f(11) = 52269
f(12) = 74588
f(13) = 103423
f(14) = 139938
f(15) = 185393
f(16) = 241144
f(17) = 308643
f(18) = 389438
f(19) = 485173
f(20) = 597588
f(21) = 728519
f(22) = 879898
f(23) = 1053753
f(24) = 1252208
f(25) = 1477483
f(26) = 1731894
f(27) = 2017853
f(28) = 2337868
f(29) = 2694543
f(30) = 3090578
f(31) = 3528769
f(32) = 4012008
f(33) = 4543283
f(34) = 5125678
f(35) = 5762373
f(36) = 6456644
f(37) = 7211863
f(38) = 8031498
f(39) = 8919113
f(40) = 9878368
```

Tarea 19:

```
TAREA 2 > ejercicio_19.py > ...
1 # Función para calcular el MCD utilizando el algoritmo de Euclides
2 def divisor():
3     # Leer los valores de primero y segundo
4     primero = int(input("Introduce el primer número: "))
5     segundo = int(input("Introduce el segundo número: "))
6
7     # Implementar el algoritmo de Euclides
8     while primero != segundo:
9         if primero > segundo:
10             primero = primero - segundo
11         else:
12             segundo = segundo - primero
13
14     # Imprimir el resultado
15     print(f"El MCD es: {primero}")
16
17 # Llamada a la función
18 divisor()
19 |
```

```
PROBLEMAS  SALIDA  TERMINAL  PUERTOS
> TERMINAL
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_19.py"
Introduce el primer número: 3
Introduce el segundo número: 4
El MCD es: 1
PS C:\Users\Usuario\Desktop\Phyton> |
```

Tarea 20:

```
TAREA 2 > ejercicio_20.py > ...
1 # Leer número decimal desde teclado
2 decimal = int(input("Introduce un número decimal: "))
3
4 # Convertir a binario usando división sucesiva
5 binario = ""
6 if decimal == 0:
7     binario = "0"
8 else:
9     while decimal > 0:
10         residuo = decimal % 2
11         binario = str(residuo) + binario
12         decimal = decimal // 2
13
14 # Mostrar el resultado
15 print("El número en binario es:", binario)
16
```

```
PROBLEMAS  SALIDA  TERMINAL  PUERTOS
> TERMINAL
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 2/ejercicio_20.py"
Introduce un número decimal: 4556
El número en binario es: 1000111001100
PS C:\Users\Usuario\Desktop\Phyton> |
```