

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

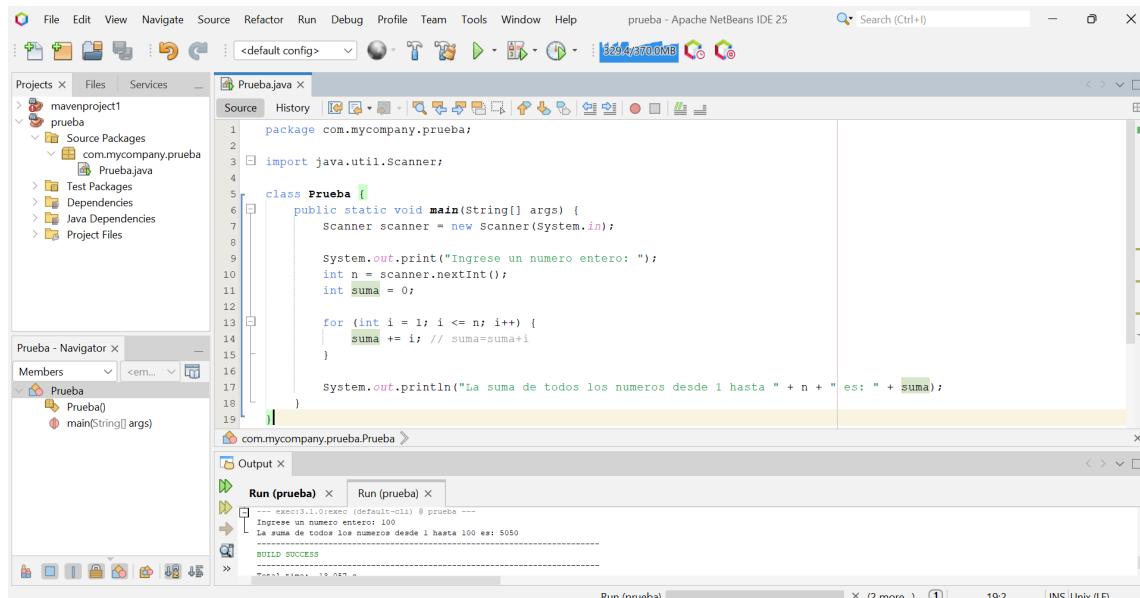
Integrantes:

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

EJERCICIOS PROPUESTOS (Java)

Elaborado por: Palacios Carrasco Roberto

1.- Dado un número entero, sacar por pantalla la suma de todos los números desde 1 hasta ese número.



```
package com.mycompany.prueba;
import java.util.Scanner;

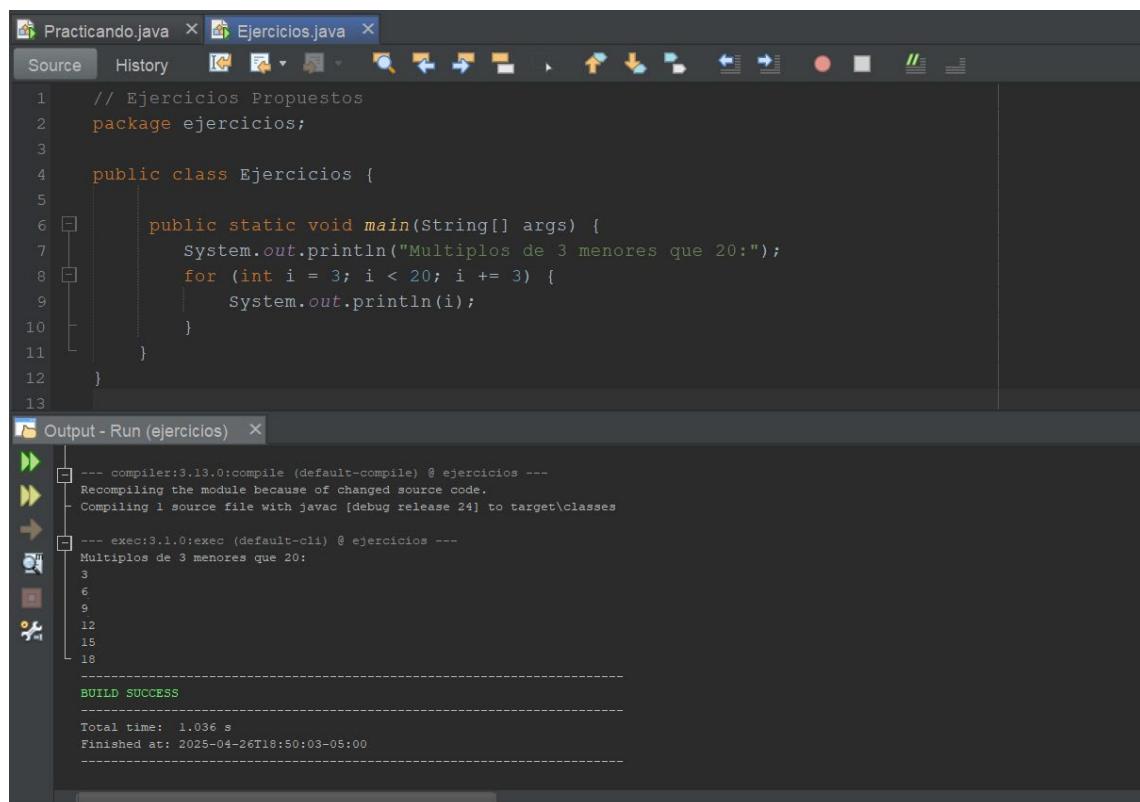
class Prueba {
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Ingrese un numero entero: ");
        int n = scanner.nextInt();
        int suma = 0;

        for (int i = 1; i <= n; i++) {
            suma += i; // suma=suma+i
        }

        System.out.println("La suma de todos los numeros desde 1 hasta " + n + " es: " + suma);
    }
}
```

2.- Escriba un programa que muestre los múltiplos de 3 menores que 20.



```
// Ejercicios Propuestos
package ejercicios;

public class Ejercicios {

    public static void main(String[] args) {
        System.out.println("Multiplos de 3 menores que 20:");
        for (int i = 3; i < 20; i += 3) {
            System.out.println(i);
        }
    }
}
```

The Output window shows the execution results:

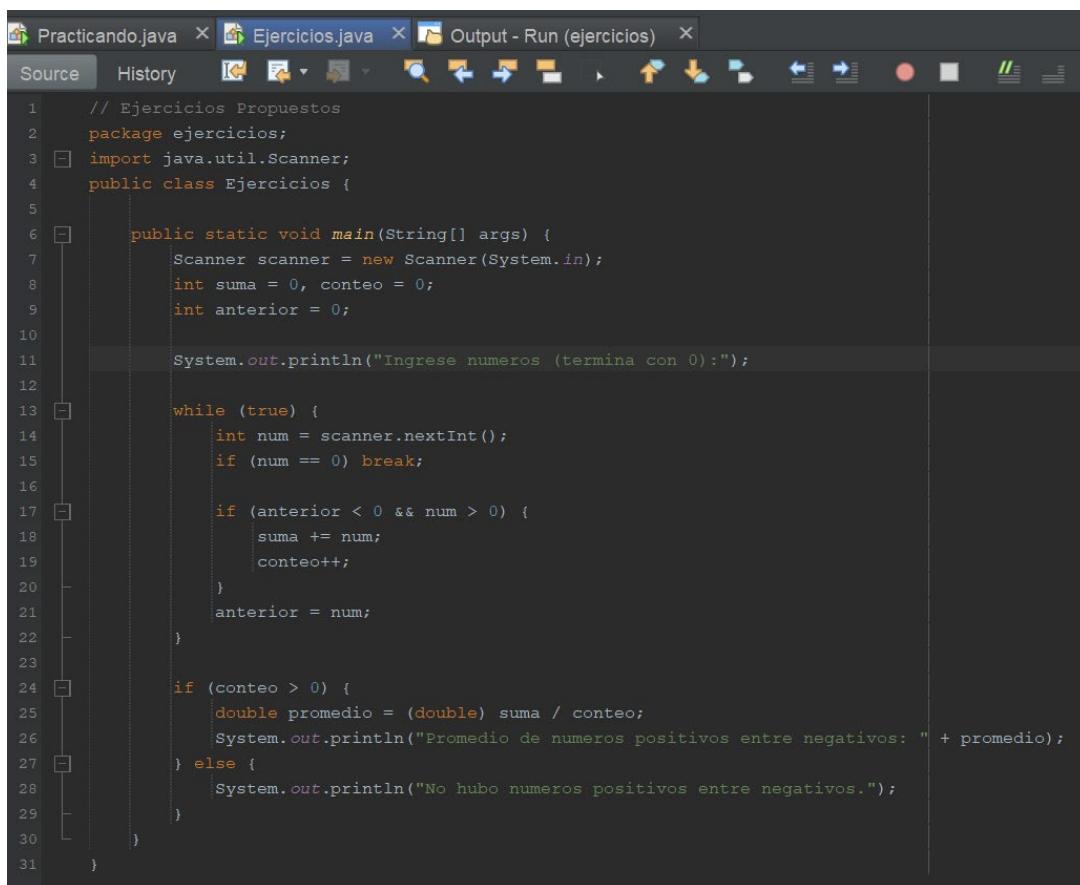
```
--- compiler:3.13.0:compile (default-compile) @ ejercicios ---
Recompiling the module because of changed source code.
Compiling 1 source file with javac [debug release 24] to target\classes

--- exec:3.1.0:exec (default-cli) @ ejercicios ---
Multiplos de 3 menores que 20:
3
6
9
12
15
18

BUILD SUCCESS

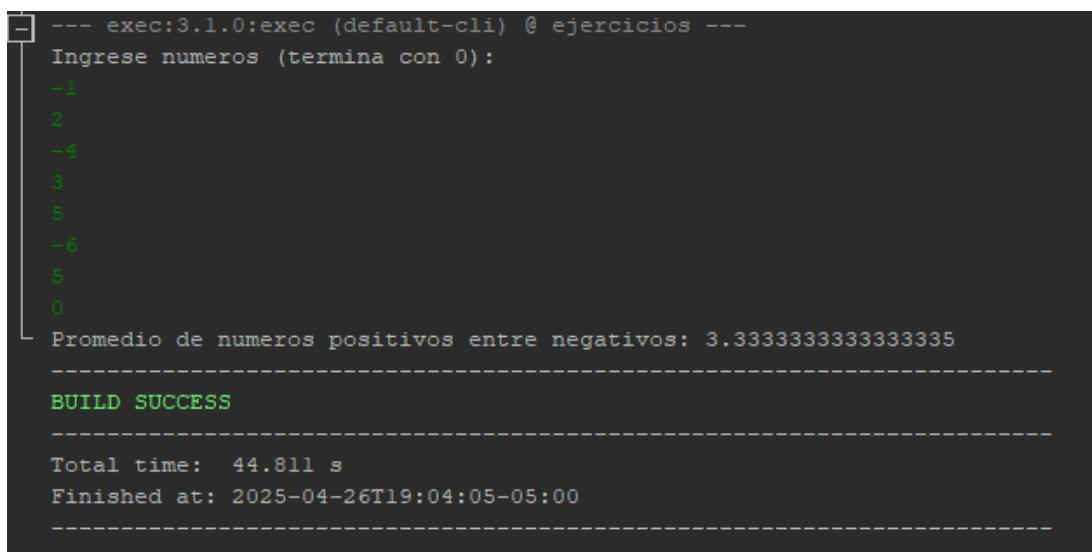
Total time: 1.036 s
Finished at: 2025-04-26T18:50:03-05:00
```

3.- Leer varios números y mostrar el promedio de los números positivos que se encuentren entre dos negativos.



The screenshot shows an IDE interface with three tabs at the top: "Practicando.java", "Ejercicios.java", and "Output - Run (ejercicios)". The "Source" tab is selected, displaying the Java code for "Ejercicios.java". The code reads numbers from standard input until it finds two consecutive negative numbers. It then calculates and prints the average of all positive numbers found between those two negatives. The code uses a Scanner to read integers and a while loop to process them.

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4 public class Ejercicios {
5
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         int suma = 0, conteo = 0;
9         int anterior = 0;
10
11         System.out.println("Ingrese numeros (termina con 0):");
12
13         while (true) {
14             int num = scanner.nextInt();
15             if (num == 0) break;
16
17             if (anterior < 0 && num > 0) {
18                 suma += num;
19                 conteo++;
20             }
21             anterior = num;
22         }
23
24         if (conteo > 0) {
25             double promedio = (double) suma / conteo;
26             System.out.println("Promedio de numeros positivos entre negativos: " + promedio);
27         } else {
28             System.out.println("No hubo numeros positivos entre negativos.");
29         }
30     }
31 }
```



The screenshot shows a terminal window with the output of running the "Ejercicios" program. It prompts the user to enter numbers, then lists several integers followed by a final "0" which serves as the termination value. The program then calculates and prints the average of the positive numbers between the first negative number (-1) and the last positive number (5). Finally, it displays a "BUILD SUCCESS" message and provides the total execution time and finish date.

```
--- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese numeros (termina con 0):
-1
2
-4
3
5
-6
5
0
Promedio de numeros positivos entre negativos: 3.3333333333333335
-----
BUILD SUCCESS
-----
Total time: 44.811 s
Finished at: 2025-04-26T19:04:05-05:00
-----
```

4.-Hacer un programa para imprimir todos los números de 1 a 10000, cuya raíz cuadrada sea igual a la suma de los dígitos de ese número. Ej. numero = 81, $8+1 = 9$.

The screenshot shows an IDE interface with two tabs: "Practicando.java" and "Ejercicios.java". The "Ejercicios.java" tab is active, displaying the following code:

```
1 // Ejercicios Propuestos
2 package ejercicios;
3
4 public class Ejercicios {
5
6     public static void main(String[] args) {
7         for (int i = 1; i <= 10000; i++) { // Iterar desde 1 hasta 10000
8             int suma = sumaDeDigitos(i);
9             if (Math.sqrt(i) == suma) {
10                 System.out.println(i);
11             }
12         }
13     }
14     // Método para calcular la suma de los dígitos de un número
15     public static int sumaDeDigitos(int num) {
16         int suma = 0;
17         while (num > 0) {
18             suma += num % 10; // Obtener el último dígito y sumarlo
19             num /= 10; // Eliminar el último dígito
20         }
21         return suma;
22     }
23 }
```

The "Output - Run (ejercicios)" tab shows the execution results:

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
1
81
BUILD SUCCESS
Total time: 1.051 s
Finished at: 2025-04-26T19:32:12-05:00
>>
```

5.- Escriba un programa que lea del teclado un valor y compruebe que se encuentra comprendido entre dos valores constantes $OPMIN$ y $OPMAX$. El programa repetirá la lectura una y otra vez hasta que el número leído cumpla esta condición. Para finalizar, lo mostrará en la pantalla, junto con el número de veces que ha necesitado repetir la operación de lectura.

The screenshot shows an IDE interface with two tabs: "Practicando.java" and "Ejercicios.java". The "Ejercicios.java" tab is active, displaying the following code:

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import ...
4 public class Ejercicios {
5
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         final int OPMIN = 10;
9         final int OPMAX = 40;
10        int intentos = 0;
11        int valor;
12
13        do {
14            System.out.print("Ingrese un valor entre " + OPMIN + " y " + OPMAX + ": ");
15            valor = scanner.nextInt();
16            intentos++;
17        } while (valor < OPMIN || valor > OPMAX);
18
19        System.out.println("Valor aceptado: " + valor);
20        System.out.println("Número de intentos: " + intentos);
21    }
22 }
```

The "Output - Run (ejercicios)" tab shows the execution results:

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese un valor entre 10 y 40: 1
Ingrese un valor entre 10 y 40: 2
Ingrese un valor entre 10 y 40: 3
Ingrese un valor entre 10 y 40: 4
Ingrese un valor entre 10 y 40: 44
Ingrese un valor entre 10 y 40: 43
Ingrese un valor entre 10 y 40: 42
Ingrese un valor entre 10 y 40: 35
Valor aceptado: 35
Número de intentos: 8
>>
```

6.- Escriba un programa que cuenta la cantidad de mellizos primos menores o iguales que un m dado. Los mellizos primos son números primos que difieren en dos unidades. Por ejemplo 3 y 5, 5 y 7, 11 y 13, etc.

```
2 package ejercicios;
3 import java.util.Scanner;
4 public class Ejercicios {
5
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print("Ingrese el valor de m: ");
10        int m = scanner.nextInt();
11        int contador = 0;
12
13        for (int i = 2; i <= m - 2; i++) {
14            if (esPrimo(i) && esPrimo(i + 2)) {
15                System.out.println("(" + i + ", " + (i + 2) + ")");
16                contador++;
17            }
18        }
19        System.out.println("Cantidad de mellizos primos: " + contador);
20    }
21
22    public static boolean esPrimo(int num) {
23        if (num < 2) return false;
24        for (int i = 2; i <= Math.sqrt(num); i++) {
25            if (num % i == 0) return false;
26        }
27        return true;
28    }
29}
```

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese el valor de m: 1000
(3, 5)
(5, 7)
(11, 13)
(17, 19)
(29, 31)
(41, 43)
(59, 61)
(71, 73)
(101, 103)
(107, 109)
(137, 139)
(149, 151)
(179, 181)
(191, 193)
(197, 199)
(227, 229)
(239, 241)
(269, 271)
(281, 283)
(311, 313)
(347, 349)
(419, 421)
(431, 433)
(461, 463)
(521, 523)
(569, 571)
(599, 601)
(617, 619)
(641, 643)
(659, 661)
(809, 811)
(821, 823)
(827, 829)
(857, 859)
(881, 883)
Cantidad de mellizos primos: 35
-----
BUILD SUCCESS
```

7.- Modifique el programa anterior para que, una vez que el juego termine (bien porque el usuario haya adivinado el número, o bien porque haya superado el número máximo de intentos), se le pregunte si desea volver a jugar. Si contesta afirmativamente, se repetirá el juego; en caso contrario, el programa finalizará.

```
-- exec:3.1.0:exec (default-cli) @ ejercicios --
Ingrese el valor de m: 10
(3, 5)
(5, 7)
Cantidad de mellizos primos: 2
Desea volver a jugar? (s/n): s
Ingrese el valor de m: 30
(3, 5)
(5, 7)
(11, 13)
(17, 19)
Cantidad de mellizos primos: 4
Desea volver a jugar? (s/n): s
Ingrese el valor de m: 55
(3, 5)
(5, 7)
(11, 13)
(17, 19)
(29, 31)
(41, 43)
Cantidad de mellizos primos: 6
Desea volver a jugar? (s/n): n
-----
BUILD SUCCESS
-----
Total time: 29.655 s
Finished at: 2025-04-26T20:45:02-05:00
```

8.- Modifique cualquiera de los dos programas anteriores de manera que tanto el número máximo como la cantidad de números en cada fila se lean desde el teclado.

The screenshot shows a Java development environment with the following details:

Source Tab: The code is located in a file named "Ejercicios.java". The code implements a program to find twin prime pairs up to a maximum value entered by the user. It includes a helper method to check if a number is prime.

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4 public class Ejercicios {
5
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8
9         System.out.print("Ingrese el valor maximo (m): ");
10        int m = scanner.nextInt();
11        System.out.print("Ingrese la cantidad de pares que desea encontrar: ");
12        int cantidadDeseada = scanner.nextInt();
13
14        int contador = 0;
15
16        for (int i = 2; i <= m - 2 && contador < cantidadDeseada; i++) {
17            if (esPrimo(i) && esPrimo(i + 2)) {
18                System.out.println("(" + i + ", " + (i + 2) + ")");
19                contador++;
20            }
21        }
22        System.out.println("Cantidad de mellizos primos encontrados: " + contador);
23    }
24
25    public static boolean esPrimo(int num) {
26        if (num < 2) return false;
27        for (int i = 2; i <= Math.sqrt(num); i++) {
28            if (num % i == 0) return false;
29        }
30        return true;
31    }
32}
```

Output Tab: The output window shows the execution of the program. The user inputs 1000 for the maximum value and 50 for the desired count of twin prime pairs. The program then lists 50 twin prime pairs found, followed by the total count of 35.

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese el valor maximo (m): 1000
Ingrese la cantidad de pares que desea encontrar: 50
(3, 5)
(5, 7)
(11, 13)
(17, 19)
(29, 31)
(41, 43)
(59, 61)
(71, 73)
(101, 103)
(107, 109)
(137, 139)
(149, 151)
(179, 181)
(191, 193)
(197, 199)
(227, 229)
(239, 241)
(269, 271)
(281, 283)
(311, 313)
(347, 349)
(419, 421)
(431, 433)
(461, 463)
(521, 523)
(569, 571)
(599, 601)
(617, 619)
(641, 643)
(659, 661)
(809, 811)
(821, 823)
(827, 829)
(857, 859)
(881, 883)
Cantidad de mellizos primos encontrados: 35
-----
BUILD SUCCESS
-----
Total time: 15.837 s
Finished at: 2025-04-26T20:52:28-05:00
```

9.- El siguiente programa calcula la potencia n-ésima positiva (x^n) de un número entero x.

```
inicio
mostrar fin_linea, "Introduzca la base y el exponente:"
leer x, n
potencia ← 1
i ← 0
mientras i<n
    potencia ← potencia * x
    i ← i+1
finMientras
mostrar fin_linea, "La potencia es ", potencia
fin
```

Modifiquelo transformando la estructura **mientras** en una estructura **para**. Luego traduzca las dos versiones del programa a un lenguaje de programación como Java o C.

The screenshot shows an IDE interface with two main panes. The top pane is titled 'Source' and contains the following Java code:

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4
5 public class Ejercicios {
6
7     public static void main(String[] args) {
8         try (Scanner scanner = new Scanner(System.in)) {
9             System.out.println("Introduzca la base y el exponente:");
10            int x = scanner.nextInt();
11            int n = scanner.nextInt();
12
13            int potenciaFor = 1;
14            for (int j = 0; j < n; j++) {
15                potenciaFor *= x;
16            }
17            System.out.println(" La potencia es " + potenciaFor);
18        }
19    }
20}
```

The bottom pane is titled 'Output - Run (ejercicios)' and shows the execution results:

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
Introduzca la base y el exponente:
2
10
La potencia es 1024
BUILD SUCCESS
Total time: 6.664 s
Finished at: 2025-04-26T22:22:14-05:00
```

16.- Construir un programa que permita multiplicar dos números enteros positivos empleando el método denominado multiplicación rusa . este método permite calcular el producto de n por m de la siguiente forma:

en pasos sucesivos se divide m por 2(división entera) y se multiplica n por 2. este proceso se repite hasta que m es 0. el resultado de la multiplicación deseada se obtiene acumulando aquellos valores sucesivos de n para los cuales el valor correspondiente de m es impar.

ejemplo: $31*27$

| N | M |
|-------|----|
| 31 * | 27 |
| 62 * | 13 |
| 124 | 6 |
| 248 * | 3 |
| 496 * | 1 |
| 992 | 0 |

si sumamos los valores marcados con asterisco(que son los que corresponden a un valor impar de m) obtenemos : $31+62+248+496=837=31*27$

The screenshot shows a Java code editor with the following code:

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4
5 public class Ejercicios {
6
7     public static void main(String[] args) {
8         try (Scanner scanner = new Scanner(System.in)) {
9             System.out.print("Ingrese el primer numero (n): ");
10            int n = scanner.nextInt();
11
12            System.out.print("Ingrese el segundo numero (m): ");
13            int m = scanner.nextInt();
14
15            int resultado = 0;
16
17            System.out.println("Proceso de multiplicación rusa:");
18            System.out.println("n\tm\tAcción");
19
20            while (m > 0) {
21                System.out.print(n + "\t" + m + "\t");
22
23                if (m % 2 != 0) {
24                    resultado += n;
25                    System.out.println("Sumar " + n + " (m es impar)");
26                } else {
27                    System.out.println("Ignorar (m es par)");
28                }
29
30                n *= 2;
31                m /= 2;
32            }
33
34            System.out.println("\nEl resultado de la multiplicacion es: " + resultado);
35        }
36    }
37}
```

```
--- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese el primer numero (n): 31
Ingrese el segundo numero (m): 27
Proceso de multiplicacion rusa:
n      m      Accion
31      27      Sumar 31 (m es impar)
62      13      Sumar 62 (m es impar)
124     6       Ignorar (m es par)
248     3       Sumar 248 (m es impar)
496     1       Sumar 496 (m es impar)

El resultado de la multiplicacion es: 837
-----
BUILD SUCCESS
-----
Total time: 7.256 s
Finished at: 2025-04-26T22:41:34-05:00
-----
```

17.- Diseña un algoritmo que encuentre todos los números primos menores que un número dado.

The screenshot shows a Java code editor with the following code:

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4
5 public class Ejercicios {
6
7     public static void main(String[] args) {
8         try (Scanner scanner = new Scanner(System.in)) {
9             System.out.print("Ingrese un numero limite: ");
10            int limite = scanner.nextInt();
11
12            System.out.println("Numeros primos menores que " + limite + ":");
13
14            for (int num = 2; num < limite; num++) {
15                if (esPrimo(num)) {
16                    System.out.print(num + " ");
17                }
18            }
19        }
20    }
21
22    private static boolean esPrimo(int numero) {
23        if (numero <= 1) {
24            return false;
25        }
26        if (numero == 2) {
27            return true;
28        }
29        if (numero % 2 == 0) {
30            return false;
31        }
32
33        for (int i = 3; i * i <= numero; i += 2) {
34            if (numero % i == 0) {
35                return false;
36            }
37        }
38
39        return true;
40    }
41 }
```

Below the code editor is the terminal output:

```
-- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese un numero limite: 150
Numeros primos menores que 150:
2 3 5 7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67 71 73 79 83 89 97 101 103 107 109 113 127 131 137 139 149
-----
BUILD SUCCESS
-----
Total time: 5.332 s
Finished at: 2025-04-26T22:49:30-05:00
```

19.- Realizar un algoritmo que escriba por pantalla el número de dígitos que tiene un valor natural leído por teclado.

```
--- exec:3.1.0:exec (default-cli) @ ejercicios ---
Ingrese un numero natural: 53465346
El numero tiene 8 digitos.

-----
BUILD SUCCESS
-----
Total time: 4.852 s
Finished at: 2025-04-26T22:51:56-05:00
-----
```

20.- Elabore un programa que imprima la figura de un triángulo rectángulo ajustada a la izquierda, formada por asteriscos. El lado del triángulo se lee como dato.

The screenshot shows an IDE interface with two panes. The left pane displays a Java code editor for a file named 'Ejercicios.java'. The code prompts the user for a triangle size and prints a right-angled triangle pattern of asterisks. The right pane shows the terminal output of running the project, including build logs and the resulting triangle pattern.

```
1 // Ejercicios Propuestos
2 package ejercicios;
3 import java.util.Scanner;
4
5 public class Ejercicios {
6
7     public static void main(String[] args) {
8         try (Scanner scanner = new Scanner(System.in)) {
9             System.out.print("Ingrese el tamaño del lado del triángulo: ");
10            int lado = scanner.nextInt();
11
12            System.out.println("Triángulo rectángulo:");
13
14            for (int i = 1; i <= lado; i++) {
15                for (int j = 1; j <= i; j++) {
16                    System.out.print("*");
17                }
18                System.out.println();
19            }
20        }
21    }
22}
```

Run (ejercicios) x Run (ejercicios) x

```
C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Documentos\NetBeansProjects\ejercicios
```

WARNING: A restricted method in java.lang.System has been called
WARNING: java.lang.System::load has been called by org.fusesource.jansi.JansiL
WARNING: Use --enable-native-access=ALL-UNNAMED to avoid a warning for callers in this
WARNING: Restricted methods will be blocked in a future release unless native access is

WARNING: A terminally deprecated method in sun.misc.Unsafe has been called
WARNING: sun.misc.Unsafe::objectFieldOffset has been called by com.google.common.util.c
WARNING: Please consider reporting this to the maintainers of class com.google.common.u
WARNING: sun.misc.Unsafe::objectFieldOffset will be removed in a future release
Scanning for projects...

```
<!-- com.mycompany:ejercicios -->
Building ejercicios 1.0-SNAPSHOT
  from pom.xml
----- [jar] -----
resources:3.3.1:resources (default-resources) @ ejercicios -->
skip non-existing resourceDirectory C:\Users\Roberto\OneDrive - Universidad Nacional Ma
compiler:3.1.0:compile (default-compile) @ ejercicios -->
Nothing to compile - all classes are up to date.

exec:3.1.0:exec (default-cli) @ ejercicios -->
Ingresé el tamaño del lado del triángulo: 8
Triángulo rectángulo:
*
**
***
****
*****
******
*****
```

BUILD SUCCESS

```
Total time: 9.658 s
Finished at: 2025-04-26T22:55:50-05:00
```

EJERCICIOS (PYTHON):

Elaborado por: Avalos Palomino Ismael Jesús

EJERCICIO 1:

1.- Dado un número entero, sacar por pantalla la suma de todos los números desde 1 hasta ese número.

```
TAREA 3 > 📁 ejercicio_1.py > ...
1   numero=int(input("introduce un numero entero:"))
2   suma=0
3   for i in range(1,numero+1):
4       suma+=i
5   print(f"suma de los numeros de 1 hasta {numero} es: {suma}")
6
```

| | | | |
|-----------|--------|----------|---------|
| PROBLEMAS | SALIDA | TERMINAL | PUERTOS |
|-----------|--------|----------|---------|

```
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 3/ejercicio _1.py"
introduce un numero entero:10
suma de los numeros de 1 hasta 10 es: 1
suma de los numeros de 1 hasta 10 es: 3
suma de los numeros de 1 hasta 10 es: 6
suma de los numeros de 1 hasta 10 es: 10
suma de los numeros de 1 hasta 10 es: 15
suma de los numeros de 1 hasta 10 es: 21
suma de los numeros de 1 hasta 10 es: 28
suma de los numeros de 1 hasta 10 es: 36
suma de los numeros de 1 hasta 10 es: 45
suma de los numeros de 1 hasta 10 es: 55
```

EJERCICIO 2:

2.- Escriba un programa que muestre los múltiplos de 3 menores que 20.

```
TAREA 3 > 📁 ejercicio _2.py > ...
1   for numero in range(1,20):
2       if numero % 3 == 0:
3           print(numero)
4
```

| | | | |
|-----------|--------|----------|---------|
| PROBLEMAS | SALIDA | TERMINAL | PUERTOS |
|-----------|--------|----------|---------|

```
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 3/ejercicio _2.py"
3
6
9
12
15
18
PS C:\Users\Usuario\Desktop\Python>
```

EJERCICIO 3:

3.- Leer varios números y mostrar el promedio de los números positivos que se encuentren entre dos negativos.

```
TAREA 3 > ➜ ejercicio_3.py > ...
1  positivos_entre_negativos = []
2  numeros = list(map(int, input("Introduce varios números separados por espacio: ").split()))
3
4  for i in range(1, len(numeros) - 1):
5      if numeros[i-1] < 0 and numeros[i] > 0 and numeros[i+1] < 0:
6          positivos_entre_negativos.append(numeros[i])
7
8  if positivos_entre_negativos:
9      promedio = sum(positivos_entre_negativos) / len(positivos_entre_negativos)
10     print(f"El promedio de los números positivos entre dos negativos es: {promedio}")
11 else:
12     print("No se encontraron números positivos entre dos negativos.")
13
```

```
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 3/ejercicio_3.py"
Introduce varios números separados por espacio: 5 -4 7 -9 2 -1 3 -6
El promedio de los números positivos entre dos negativos es: 4.0
PS C:\Users\Usuario\Desktop\Python>
```

EJERCICIO 4:

4.-Hacer un programa para imprimir todos los números de 1 a 10000, cuya raíz cuadrada sea igual a la suma de los dígitos de ese número. Ej. numero = 81, $8+1 = 9$.

```
TAREA 3 > ➜ ejercicio_4.py > ...
1  import math
2  for numero in range(1, 10001):
3      raiz_cuadrada = math.sqrt(numero)
4      suma_digitos = sum(int(dígito) for dígito in str(numero))
5
6      if raiz_cuadrada == suma_digitos and raiz_cuadrada * raiz_cuadrada == numero:
7          print(numero)
8
```

```
PROBLEMAS SALIDA TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 3/ejercicio_4.py"
1
81
PS C:\Users\Usuario\Desktop\Python>
```

EJERCICIO 5:

5.- Escriba un programa que lea del teclado un valor y compruebe que se encuentra comprendido entre dos valores constantes $OPMIN$ y $OPMAX$. El programa repetirá la lectura una y otra vez hasta que el número leído cumpla esta condición. Para finalizar, lo mostrará en la pantalla, junto con el número de veces que ha necesitado repetir la operación de lectura.

```
TAREA 3 > ejercicio_5.py > ...
1  OPMIN = 10
2  OPMAX = 100
3  # Inicializamos el contador
4  intentos = 0
5  while True:
6      valor = int(input(f"Introduce un número entre {OPMIN} y {OPMAX}: "))
7      intentos += 1
8      if OPMIN <= valor <= OPMAX:
9          break
10
11 print(f"\nNúmero válido ingresado: {valor}")
12 print(f"Número de intentos: {intentos}")
13
```

PROBLEMAS SALIDA TERMINAL PUERTOS

```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 3/ejercicio _5.py"
Introduce un numero entre 10 y 100: 50
Número válido ingresado: 50
Número de intentos: 1
PS C:\users\Usuario\Desktop\Phyton> █
```

EJERCICIO 6:

6.- Escriba un programa que cuenta la cantidad de mellizos primos menores o iguales que un m dado. Los mellizos primos son números primos que difieren en dos unidades. Por ejemplo 3 y 5, 5 y 7, 11 y 13, etc.

```
TAREA 3 > ejercicio_6.py > ...
1  def es_primo(n):
2      if n < 2:
3          return False
4      for i in range(2, int(n**0.5) + 1):
5          if n % i == 0:
6              return False
7      return True
8  m = int(input("Introduce el valor de m: "))
9  contador = 0
10 for i in range(2, m - 1):
11     if es_primo(i) and es_primo(i + 2):
12         if i + 2 <= m:
13             contador += 1
14         print(f"Par de mellizos primos: ({i}, {i+2})")
15
16 print(f"\nCantidad de pares de mellizos primos menores o iguales que {m}: {contador}")
17
```

PROBLEMAS SALIDA TERMINAL PUERTOS

```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/ejercicio_6.py
Introduce el valor de m: 54
Par de mellizos primos: (3, 5)
Par de mellizos primos: (5, 7)
Par de mellizos primos: (11, 13)
Par de mellizos primos: (17, 19)
Par de mellizos primos: (29, 31)
Par de mellizos primos: (41, 43)
```

```
Cantidad de pares de mellizos primos menores o iguales que 54: 6
PS C:\Users\Usuario\Desktop\Phyton>
```

EJERCICIO 7:

7.- Modifique el programa anterior para que, una vez que el juego termine (bien porque el usuario haya adivinado el número, o bien porque haya superado el número máximo de intentos), se le pregunte si desea volver a jugar. Si contesta afirmativamente, se repetirá el juego; en caso contrario, el programa finalizará.

```
TAREA 3 > ejercicio_7.py > ...
1  import random
2
3  def jugar():
4      numero_secreto = random.randint(1, 100)
5      intentos_maximos = 5
6      intentos = 0
7
8      print("¡Adivina el número entre 1 y 100!")
9
10     while intentos < intentos_maximos:
11         try:
12             intento = int(input("Introduce tu número: "))
13         except ValueError:
14             print("Por favor, introduce un número válido.")
15             continue
16
17         intentos += 1
18
19         if intento == numero_secreto:
20             print(f"¡Felicitaciones! Adivinaste el número en {intentos} intentos.")
21             break
22         elif intento < numero_secreto:
23             print("El número es mayor.")
24         else:
25             print("El número es menor.")
26
27     else:
28         print(f"Lo siento, has agotado tus {intentos_maximos} intentos. El número era {numero_secreto}.")
29
30 def main():
31     while True:
32         jugar()
33         respuesta = input("¿Quieres jugar otra vez? (s/n): ").strip().lower()
34         if respuesta != 's':
35             print("¡Gracias por jugar! ¡Hasta la próxima!")
36             break
37
38 if __name__ == "__main__":
39     main()
```

PROBLEMAS

SALIDA

TERMINAL

PUERTOS

```
PS C:\Users\Usuario\Desktop\Python> & c:
¡Adivina el número entre 1 y 100!
Introduce tu número: 59
El número es menor.
Introduce tu número: 
```

EJERCICIO 8:

8.- Modifique cualquiera de los dos programas anteriores de manera que tanto el número máximo como la cantidad de números en cada fila se lean desde el teclado.

```
TAREA 3 > ejercicio_8.py > jugar
 1 import random
 2 def jugar(numero_maximo, numeros_por_fila):
 3     numero_secreto = random.randint(1, numero_maximo)
 4     intentos_maximos = 5
 5     intentos = 0
 6     print("\nAdivina el número entre 1 y {numero_maximo}!")
 7     print("Aquí están los números posibles:")
 8     for i in range(1, numero_maximo + 1):
 9         print(f"[{i}]", end=" ")
10     if i % numeros_por_fila == 0:
11         print() # Salto de línea cada "numeros_por_fila"
12     print() # Línea en blanco después de mostrar todos los números
13     while intentos < intentos_maximos:
14         try:
15             intento = int(input("\nIntroduce tu número: "))
16         except ValueError:
17             print("Por favor, introduce un número válido.")
18             continue
19         if intento < 1 or intento > numero_maximo:
20             print(f"El número debe estar entre 1 y {numero_maximo}.")
21             continue
22         intentos += 1
23         if intento == numero_secreto:
24             print(f"¡Felicitaciones! Adivinaste el número en {intentos} intentos.")
25             break
26         elif intento < numero_secreto:
27             print("El número secreto es mayor.")
28         else:
29             print("El número secreto es menor.")
30         else:
31             print(f"Lo siento, has agotado tus {intentos_maximos} intentos. El número era {numero_secreto}.")
32     def main():
33         while True:
34             try:
35                 numero_maximo = int(input("\nIngrese el número máximo (ej: 100): "))
36                 numeros_por_fila = int(input("Ingrese la cantidad de números por fila al mostrar los posibles: "))
37             except ValueError:
38                 print("Por favor, ingrese números válidos.")
39                 continue
40             jugar(numero_maximo, numeros_por_fila)
41             respuesta = input("\n¿Quieres jugar otra vez? (s/n): ").strip().lower()
42             if respuesta != 's':
43                 print("¡Gracias por jugar! ¡Hasta la próxima!")
44                 break
45     if __name__ == "__main__":
46         main()
```

```
PROBLEMAS SALIDA TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 3/ejercicio_8.py"
Ingrese el número máximo (ej: 100): 100
Ingrese la cantidad de números por fila al mostrar los posibles: 54
¡Adivina el número entre 1 y 100!
Aquí están los números posibles:
 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50
 51 52 53 54
 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 100

Introduce tu número: 28
El número secreto es mayor.

Introduce tu número: 45
El número secreto es mayor.

Introduce tu número: 54
El número secreto es mayor.

Introduce tu número: 53
El número secreto es mayor.

Introduce tu número: 52
El número secreto es mayor.

Introduce tu número: 51
El número secreto es mayor.

Introduce tu número: 50
El número secreto es mayor.

Introduce tu número: 49
El número secreto es mayor.

Introduce tu número: 48
El número secreto es mayor.

Introduce tu número: 47
El número secreto es mayor.

Introduce tu número: 46
El número secreto es mayor.

Introduce tu número: 45
El número secreto es mayor.

Introduce tu número: 44
El número secreto es mayor.

Introduce tu número: 43
El número secreto es mayor.

Introduce tu número: 42
El número secreto es mayor.

Introduce tu número: 41
El número secreto es mayor.

Introduce tu número: 40
El número secreto es mayor.

Introduce tu número: 39
El número secreto es mayor.

Introduce tu número: 38
El número secreto es mayor.

Introduce tu número: 37
El número secreto es mayor.

Introduce tu número: 36
El número secreto es mayor.

Introduce tu número: 35
El número secreto es mayor.

Introduce tu número: 34
El número secreto es mayor.

Introduce tu número: 33
El número secreto es mayor.

Introduce tu número: 32
El número secreto es mayor.

Introduce tu número: 31
El número secreto es mayor.

Introduce tu número: 30
El número secreto es mayor.

Introduce tu número: 29
El número secreto es mayor.

Introduce tu número: 28
El número secreto es mayor.

Introduce tu número: 27
El número secreto es mayor.

Introduce tu número: 26
El número secreto es mayor.

Introduce tu número: 25
El número secreto es mayor.

Introduce tu número: 24
El número secreto es mayor.

Introduce tu número: 23
El número secreto es mayor.

Introduce tu número: 22
El número secreto es mayor.

Introduce tu número: 21
El número secreto es mayor.

Introduce tu número: 20
El número secreto es mayor.

Introduce tu número: 19
El número secreto es mayor.

Introduce tu número: 18
El número secreto es mayor.

Introduce tu número: 17
El número secreto es mayor.

Introduce tu número: 16
El número secreto es mayor.

Introduce tu número: 15
El número secreto es mayor.

Introduce tu número: 14
El número secreto es mayor.

Introduce tu número: 13
El número secreto es mayor.

Introduce tu número: 12
El número secreto es mayor.

Introduce tu número: 11
El número secreto es mayor.

Introduce tu número: 10
El número secreto es mayor.

Introduce tu número: 9
El número secreto es mayor.

Introduce tu número: 8
El número secreto es mayor.

Introduce tu número: 7
El número secreto es mayor.

Introduce tu número: 6
El número secreto es mayor.

Introduce tu número: 5
El número secreto es mayor.

Introduce tu número: 4
El número secreto es mayor.

Introduce tu número: 3
El número secreto es mayor.

Introduce tu número: 2
El número secreto es mayor.

Introduce tu número: 1
El número secreto es mayor.
```

EJERCICIO 9:

9.- El siguiente programa calcula la potencia n-ésima positiva (x^n) de un número entero x .

```
inicio
mostrar fin_linea, "Introduzca la base y el exponente:"
leer x, n
potencia ← 1
i ← 0
mientras i<n
    potencia ← potencia * x
    i ← i+1
finMientras
mostrar fin_linea, "La potencia es ", potencia
fin
```

Modifíquelo transformando la estructura **mientras** en una estructura **para**. Luego traduzca las dos versiones del programa a un lenguaje de programación como Java o C.

```
TAREA 3 > ⚙ ejercicio_9.py > ...
1  print("Introduzca la base y el exponente:")
2  x = int(input("Base: "))
3  n = int(input("Exponente: "))
4
5  potencia = 1
6
7  for i in range(n):
8      potencia = potencia * x
9
10 print("La potencia es", potencia)
```

| | | | |
|---|--------|-----------------|---------|
| PROBLEMAS | SALIDA | <u>TERMINAL</u> | PUERTOS |
| PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/!< Introduzca la base y el exponente: Base: 54 Exponente: 5 La potencia es 459165024 | | | |

EJERCICIO 10:

10.- Desarrolle un programa que determine la paga bruta de cada uno de varios empleados. La compañía paga “tiempo normal” por las primeras 40 horas laboradas por cada empleado y paga “tiempo y medio” por todas las horas laboradas en exceso de 40 horas. Usted recibe una lista de los empleados de la compañía, el número de horas que cada empleado trabajó la semana anterior y el sueldo por hora de cada empleado. Su programa deberá introducir esta información para cada empleado y determinar y exhibir la paga bruta del empleado.

Por ejemplo (el texto en negrilla corresponde a los datos introducidos por el usuario):

```
Teclee horas laboradas (0 para terminar): 39
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 390
Teclee horas laboradas (0 para terminar): 40
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 400
Teclee horas laboradas (0 para terminar): 41
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 415
```

```
TAREA 3 > ➜ ejercicio_10.py > ...
1  while True:
2      horas = float(input("Teclee horas laboradas (0 para terminar): "))
3
4      if horas == 0:
5          break
6
7      sueldo_por_hora = float(input("Teclee sueldo por hora del trabajador $: "))
8
9      if horas <= 40:
10         paga = horas * sueldo_por_hora
11     else:
12         paga = 40 * sueldo_por_hora
13         horas_extra = horas - 40
14         paga += horas_extra * sueldo_por_hora * 1.5
15
16     print(f"La paga es $: {paga:.2f}\n")
17
```

| | | | |
|-----------|--------|--|---------|
| PROBLEMAS | SALIDA | <u>TERMINAL</u> | PUERTOS |
| | | <pre>PS C:\Users\Usuario\Desktop\Python> & c:/Users/I Teclee horas laboradas (0 para terminar): 14 Teclee sueldo por hora del trabajador \$: 45 La paga es \$: 630.00</pre> | |

EJERCICIO 11:

11.- El número 73331 tiene la fascinante propiedad de que no solo es primo sino que al ir eliminando (suprimiendo) el dígito menos significativo, también sigue siendo un número primo. Es decir: 7333, 733, 73, 7, también son primos. Dado un número verifique si cumple tal propiedad mediante un programa.

```
TAREA 3 > ejercicio_11.py > es_primo
1  def es_primo(n):
2      if n < 2:
3          return False
4      for i in range(2, int(n**0.5) + 1):
5          if n % i == 0:
6              return False
7      return True
8
9  def verificar_propiedad(numero):
10     while numero > 0:
11         if not es_primo(numero):
12             return False
13         numero //= 10
14     return True
15
16 numero = int(input("Ingrese un número para verificar: "))
17
18 if verificar_propiedad(numero):
19     print(f"El número {numero} cumple la propiedad.")
20 else:
21     print(f"El número {numero} NO cumple la propiedad.")
```

```
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 3/ejercicio_11.py"
Ingrese un numero para verificar: 5
El número 5 cumple la propiedad.
```

EJERCICIO 12:

12.- Leer una serie de números distintos de cero, el ultimo numero de la serie es -99 y obtener el numero mayor. Como resultado se debe visualizar el numero mayor y un mensaje de indicación de numero negativo, en caso de que se haya elido un numero negativo

```
TAREA 3 > ejercicios_12.py > ...
1 mayor = None
2 hubo_negativo = False
3 print("Ingrese números (ingrese -99 para terminar):")
4 while True:
5     numero = int(input("Número: "))
6     if numero == -99:
7         break
8     if numero == 0:
9         print("Cero no es un valor válido. Intente nuevamente.")
10        continue
11     if mayor is None or numero > mayor:
12         mayor = numero
13     if numero < 0:
14         hubo_negativo = True
15     if mayor is not None:
16         print(f"\nEl número mayor es: {mayor}")
17         if hubo_negativo:
18             print("Se ingresaron números negativos.")
19     else:
20         print("\nNo se ingresaron números válidos.")
```

PROBLEMAS SALIDA TERMINAL PUERTOS

```
PS C:\Users\Usuario\Desktop\Python> & c:/Users,
Ingrese números (ingrese -99 para terminar):
Número: 49494
Número: 55454
Número: 626
Número: 66565
Número: 565
Número: -656
Número: -999
Número: -99
```

```
El número mayor es: 66565
Se ingresaron números negativos.
PS C:\Users\Usuario\Desktop\Python>
```

EJERCICIO 13:

13.- Calcular el valor de x dada n en la siguiente ecuación:
 $x = \frac{1}{2}^2 + \frac{2}{3}^3 + \frac{3}{4}^4 + \frac{4}{5}^5 \dots n-1/n^n$

```
TAREA 3 > 📁 ejercicio_13.py > ...
1  n = int(input("Ingrese el valor de n: "))
2  x = 0
3  for i in range(2, n):
4      x += i / (i + 1)
5  print(f"\nEl valor de x es: {x:.4f}")
6  |
```

PROBLEMAS SALIDA TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Phyton> & c:/
Ingrese el valor de n: 54

El valor de x es: 48.9246

PS C:\Users\Usuario\Desktop\Phyton> █

EJERCICIO 14:

14.- La capacidad de un deposito es de V litros. mediante una valvula se introducen LE litros por segundo en un deposito; mediante un desague, se extrae LS litros por segundo. si el deposito tiene inicialmente 10 litros , diseñar un programa que muestre, segundo a segundo, durante un minuto, el volumen de agua almacenada. Al final debe indicar si se rebasara la capacidad de almacenamiento del deposito?, si permanecera siempre con la misma cantidad de agua? o si se vaciara?.

```
TAREA 3 > ejercicio_14.py > ...
1  V = float(input("Ingrese la capacidad del depósito (litros): "))
2  LE = float(input("Ingrese los litros que entran por segundo (LE): "))
3  LS = float(input("Ingrese los litros que salen por segundo (LS): "))
4  l0 = float(input("Ingrese el volumen inicial de agua (litros): "))
5  volumen = l0
6  rebasado = False
7  vacio = False
8  constante = True
9  print("\n--- Volumen segundo a segundo ---")
10 for segundo in range(1, 61):
11     volumen = volumen + LE - LS
12
13     if volumen > V:
14         volumen = V
15         rebasado = True
16     if volumen <= 0:
17         volumen = 0
18         vacio = True
19
20     if LE != LS:
21         constante = False
22
23     print(f"Segundo {segundo}: {volumen:.2f} litros")
24 print("\n--- Resultado Final ---")
25 if rebasado:
26     print("¡Se rebasó la capacidad del depósito!")
27 elif constante:
28     print("El volumen de agua se mantuvo constante todo el tiempo.")
29 elif vacio:
30     print("¡El depósito se vació en algún momento!")
31 else:
32     print("El volumen de agua cambió, pero no se desbordó ni se vació.")
33
```

```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/
Ingrese la capacidad del depósito (litros): 5
Ingrese los litros que entran por segundo (LE): 4
Ingrese los litros que salen por segundo (LS): 2
Ingrese el volumen inicial de agua (litros): 1

--- Volumen segundo a segundo ---
Segundo 1: 3.00 litros
Segundo 2: 5.00 litros
Segundo 3: 5.00 litros
Segundo 4: 5.00 litros
Segundo 5: 5.00 litros
Segundo 6: 5.00 litros
Segundo 7: 5.00 litros
Segundo 8: 5.00 litros
Segundo 9: 5.00 litros
Segundo 10: 5.00 litros
Segundo 11: 5.00 litros
Segundo 12: 5.00 litros
Segundo 13: 5.00 litros
Segundo 14: 5.00 litros
Segundo 15: 5.00 litros
Segundo 16: 5.00 litros
Segundo 17: 5.00 litros
Segundo 18: 5.00 litros
Segundo 19: 5.00 litros
Segundo 20: 5.00 litros
Segundo 21: 5.00 litros
Segundo 22: 5.00 litros
Segundo 23: 5.00 litros
Segundo 24: 5.00 litros
Segundo 25: 5.00 litros
Segundo 26: 5.00 litros
Segundo 27: 5.00 litros
Segundo 28: 5.00 litros
Segundo 29: 5.00 litros
Segundo 30: 5.00 litros
Segundo 31: 5.00 litros
Segundo 32: 5.00 litros
Segundo 33: 5.00 litros
Segundo 34: 5.00 litros
```

Segundo 23: 5.00 litros
Segundo 24: 5.00 litros
Segundo 25: 5.00 litros
Segundo 26: 5.00 litros
Segundo 27: 5.00 litros
Segundo 28: 5.00 litros
Segundo 29: 5.00 litros
Segundo 30: 5.00 litros
Segundo 31: 5.00 litros
Segundo 32: 5.00 litros
Segundo 33: 5.00 litros
Segundo 34: 5.00 litros
Segundo 35: 5.00 litros
Segundo 36: 5.00 litros
Segundo 37: 5.00 litros
Segundo 38: 5.00 litros
Segundo 39: 5.00 litros
Segundo 40: 5.00 litros
Segundo 41: 5.00 litros
Segundo 42: 5.00 litros
Segundo 43: 5.00 litros
Segundo 44: 5.00 litros
Segundo 45: 5.00 litros
Segundo 46: 5.00 litros
Segundo 47: 5.00 litros
Segundo 48: 5.00 litros
Segundo 49: 5.00 litros
Segundo 50: 5.00 litros
Segundo 51: 5.00 litros
Segundo 52: 5.00 litros
Segundo 53: 5.00 litros
Segundo 54: 5.00 litros
Segundo 55: 5.00 litros
Segundo 56: 5.00 litros
Segundo 57: 5.00 litros
Segundo 58: 5.00 litros
Segundo 59: 5.00 litros
Segundo 60: 5.00 litros

--- Resultado Final ---

¡Se rebasó la capacidad del depósito!

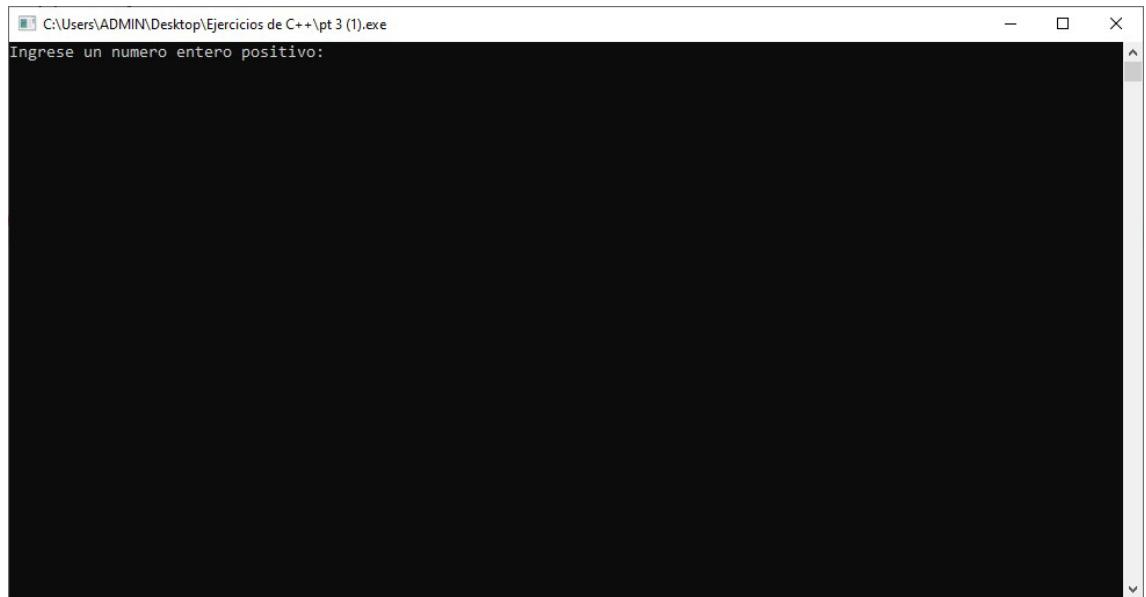
EJERCICIOS DE (C++):

Elaborado por: Melendez Morales Santiago

EJERCICIO 1:

1.- Dado un número entero, sacar por pantalla la suma de todos los números desde 1 hasta ese número.

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



EJERCICIO 2:

2.- Escriba un programa que muestre los múltiplos de 3 menores que 20.

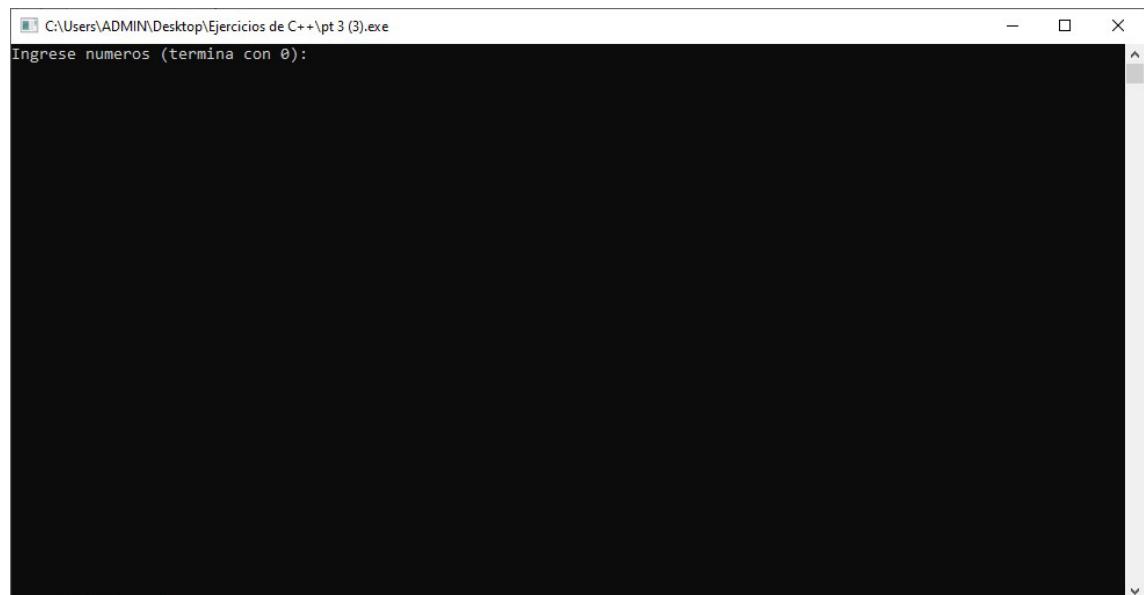
```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     cout << "Multiplos de 3 menores que 20:" << endl;
6
7     for (int i = 1; i < 20; i++) {
8         if (i % 3 == 0) {
9             cout << i << endl;
10        }
11    }
12
13    return 0;
14 }
```

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (2).exe
Multiplos de 3 menores que 20:
3
6
9
12
15
18
-----
Process exited after 0.03619 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 3:

3.- Leer varios números y mostrar el promedio de los números positivos que se encuentren entre dos negativos.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int numero;
6     int suma = 0, contador = 0;
7     bool primerNegativoEncontrado = false;
8
9     cout << "Ingrese numeros (termina con 0):" << endl;
10
11    while (true) {
12        cin >> numero;
13
14        if (numero == 0) {
15            break; // Terminamos si el usuario ingresa 0
16        }
17
18        if (numero < 0) {
19            if (!primerNegativoEncontrado) {
20                // Primer negativo encontrado, ahora empezamos a contar positivos
21                primerNegativoEncontrado = true;
22            } else {
23                // Segundo negativo encontrado, terminamos de leer
24                break;
25            }
26        } else {
27            if (primerNegativoEncontrado) {
28                // Solo contamos positivos DESPUÉS del primer negativo
29                suma += numero;
30                contador++;
31            }
32        }
33    }
34
35    if (contador > 0) {
36        double promedio = static_cast<double>(suma) / contador;
37        cout << "El promedio de los numeros positivos entre dos negativos es: " << promedio << endl;
38    } else {
39        cout << "No hubo numeros positivos entre dos negativos." << endl;
40    }
41
42    return 0;
43 }
```



EJERCICIO 4:

4.-Hacer un programa para imprimir todos los números de 1 a 10000, cuya raíz cuadrada sea igual a la suma de los dígitos de ese número. Ej. numero = 81, $8+1 = 9$.

```
1 #include <iostream>
2 #include <cmath> // Para usar sqrt
3 using namespace std;
4
5 // Función para sumar los dígitos de un número
6 int sumaDigitos(int numero) {
7     int suma = 0;
8     while (numero > 0) {
9         suma += numero % 10;
10        numero /= 10;
11    }
12    return suma;
13}
14
15 int main() {
16     cout << "Numeros entre 1 y 10000 cuya raiz cuadrada es igual a la suma de sus digitos:" << endl;
17
18     for (int i = 1; i <= 10000; i++) {
19         int raiz = sqrt(i);
20         if (raiz * raiz == i) { // Comprobar que sea un cuadrado perfecto
21             if (sumaDigitos(i) == raiz) {
22                 cout << i << endl;
23             }
24         }
25     }
26
27     return 0;
28 }
```

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt3 (4).exe
Numeros entre 1 y 10000 cuya raiz cuadrada es igual a la suma de sus digitos:
1
81

-----
Process exited after 0.4704 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 5:

5.- Escriba un programa que lea del teclado un valor y compruebe que se encuentra comprendido entre dos valores constantes *OPMIN* y *OPMAX*. El programa repetirá la lectura una y otra vez hasta que el número leído cumpla esta condición. Para finalizar, lo mostrará en la pantalla, junto con el número de veces que ha necesitado repetir la operación de lectura.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     const int OPMIN = 10; // Puedes cambiar estos valores si quieres
6     const int OPMAX = 50;
7     int valor;
8     int intentos = 0;
9
10    do {
11        cout << "Ingrese un valor entre " << OPMIN << " y " << OPMAX << ": ";
12        cin >> valor;
13        intentos++;
14    } while (valor < OPMIN || valor > OPMAX);
15
16    cout << "Valor correcto: " << valor << endl;
17    cout << "Cantidad de intentos: " << intentos << endl;
18
19    return 0;
20 }
```

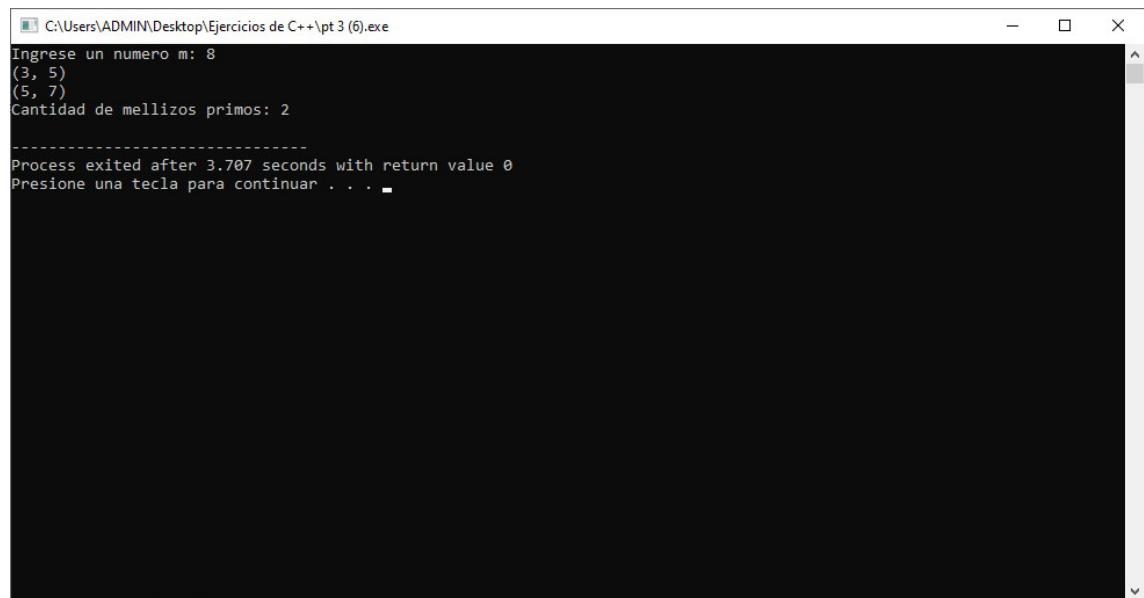
```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (5).exe
Ingrese un valor entre 10 y 50: 30
Valor correcto: 30
Cantidad de intentos: 1

-----
Process exited after 4.494 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 6:

6.- Escriba un programa que cuenta la cantidad de mellizos primos menores o iguales que un m dado. Los mellizos primos son números primos que difieren en dos unidades. Por ejemplo 3 y 5, 5 y 7, 11 y 13, etc.

```
1 #include <iostream>
2 using namespace std;
3
4 // Función para verificar si un número es primo
5 bool esPrimo(int n) {
6     if (n <= 1) return false;
7     for (int i = 2; i * i <= n; i++) {
8         if (n % i == 0) return false;
9     }
10    return true;
11 }
12
13 int main() {
14     int m, contador = 0;
15
16     cout << "Ingrese un numero m: ";
17     cin >> m;
18
19     for (int i = 2; i <= m - 2; i++) { // Hasta m-2 para que i+2 no se pase
20         if (esPrimo(i) && esPrimo(i + 2)) {
21             cout << "(" << i << ", " << i + 2 << ")" << endl;
22             contador++;
23         }
24     }
25
26     cout << "Cantidad de mellizos primos: " << contador << endl;
27
28     return 0;
29 }
```



The screenshot shows a terminal window with the following output:

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (6).exe
Ingrese un numero m: 8
(3, 5)
(5, 7)
Cantidad de mellizos primos: 2

-----
Process exited after 3.707 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 7:

7.- Modifique el programa anterior para que, una vez que el juego termine (bien porque el usuario haya adivinado el número, o bien porque haya superado el número máximo de intentos), se le pregunte si desea volver a jugar. Si contesta afirmativamente, se repetirá el juego; en caso contrario, el programa finalizará.

```
1 #include <iostream>
2 #include <cstdlib> // Para rand() y srand()
3 #include <ctime>   // Para time()
4 using namespace std;
5
6 int main() {
7     int numeroSecreto, intento, intentosMaximos = 5;
8     char respuesta;
9
10    srand(time(0)); // Inicializar la semilla aleatoria
11
12    do {
13        numeroSecreto = rand() % 100 + 1; // Número secreto entre 1 y 100
14        int intentos = 0;
15        bool adivinado = false;
16
17        cout << "Adivina el numero secreto entre 1 y 100!" << endl;
18
19        while (intentos < intentosMaximos) {
20            cout << "Intento " << (intento + 1) << " de " << intentosMaximos << ":" ;
21            cin >> intento;
22            intentos++;
23
24            if (intento == numeroSecreto) {
25                cout << ";Felicitaciones! Adivinaste el numero en " << intentos << " intento(s)." << endl;
26                adivinado = true;
27                break;
28            } else if (intento < numeroSecreto) {
29                cout << "El numero secreto es mayor." << endl;
30            } else {
31                cout << "El numero secreto es menor." << endl;
32            }
33        }
34
35        if (!adivinado) {
36            cout << "Lo siento, se acabaron los intentos. El numero secreto era: " << numeroSecreto << endl;
37        }
38
39        cout << "¿Deseas jugar de nuevo? (s/n): ";
40        cin >> respuesta;
41
42    } while (respuesta == 's' || respuesta == 'S');
43
44    cout << ";Gracias por jugar!" << endl;
45
46    return 0;
47 }
```

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (7).exe
Adivina el numero secreto entre 1 y 100!
Intento 1 de 5: 7
El numero secreto es menor.
Intento 2 de 5: 6
El numero secreto es menor.
Intento 3 de 5: 3
;Felicitaciones! Adivinaste el numero en 3 intento(s).
;Deseas jugar de nuevo? (s/n):
```

EJERCICIO 8:

8.- Modifique cualquiera de los dos programas anteriores de manera que tanto el número máximo como la cantidad de números en cada fila se lean desde el teclado.

```
1 #include <iostream>
2 #include <cstdlib> // Para rand() y srand()
3 #include <ctime> // Para time()
4 using namespace std;
5
6 int main() {
7     int numeroSecreto, intento, numeroMaximo, intentosMaximos;
8     char respuesta;
9
10    srand(time(0)); // Inicializar la semilla aleatoria
11
12    do {
13        // Leer desde teclado los valores
14        cout << "Ingrese el numero maximo que puede tener el numero secreto: ";
15        cin >> numeroMaximo;
16
17        cout << "Ingrese la cantidad de intentos permitidos: ";
18        cin >> intentosMaximos;
19
20        numeroSecreto = rand() % numeroMaximo + 1; // Número secreto entre 1 y numeroMaximo
21        int intentos = 0;
22        bool adivinado = false;
23
24        cout << "\nAdivina el numero secreto entre 1 y " << numeroMaximo << "!" << endl;
25
26        while (intentos < intentosMaximos) {
27            cout << "Intento " << (intentos + 1) << " de " << intentosMaximos << ":" ;
28            cin >> intento;
29            intentos++;
30
31            if (intento == numeroSecreto) {
32                cout << "¡Felicitaciones! Adivinaste el numero en " << intentos << " intento(s)." << endl;
33                adivinado = true;
34                break;
35            } else if (intento < numeroSecreto) {
36                cout << "El numero secreto es mayor." << endl;
37            } else {
38                cout << "El numero secreto es menor." << endl;
39            }
40        }
41
42        if (!adivinado) {
43            cout << "Lo siento, se acabaron los intentos. El numero secreto era: " << numeroSecreto << endl;
44        }
45
46        cout << "¿Deseas jugar de nuevo? (s/n): ";
47        cin >> respuesta;
48        cout << endl;
49
50    } while (respuesta == 's' || respuesta == 'S');
51
52    cout << "¡Gracias por jugar!" << endl;
53
54    return 0;
55 }
```

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (8).exe
Ingrese el numero maximo que puede tener el numero secreto: 100
Ingrese la cantidad de intentos permitidos: 50

TíAdivina el numero secreto entre 1 y 100!
Intento 1 de 50: 50
El numero secreto es mayor.
Intento 2 de 50: 80
El numero secreto es menor.
Intento 3 de 50: 60
El numero secreto es mayor.
Intento 4 de 50: 70
El numero secreto es mayor.
Intento 5 de 50: 75
El numero secreto es menor.
Intento 6 de 50: 72
El numero secreto es mayor.
Intento 7 de 50: 73
El numero secreto es mayor.
Intento 8 de 50: 74
TíFelicitas! Adivinaste el numero en 8 intento(s).
TíDeseas jugar de nuevo? (s/n): -
```

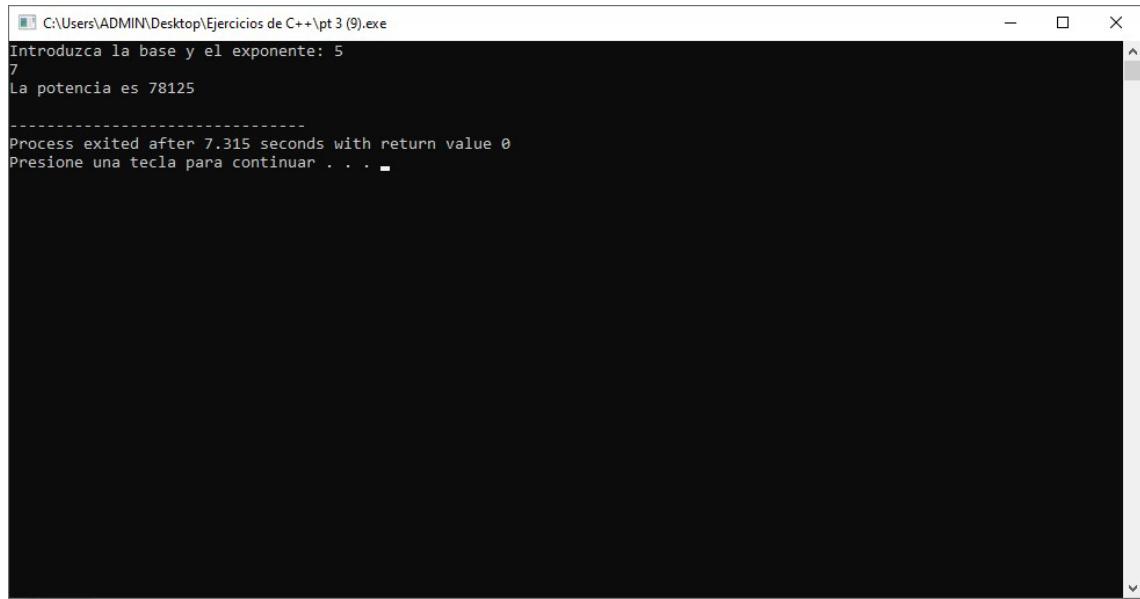
EJERCICIO 9:

9.- El siguiente programa calcula la potencia n-ésima positiva (x^n) de un número entero x .

```
inicio
mostrar fin_linea, "Introduzca la base y el exponente:"
leer x, n
potencia ← 1
i ← 0
mientras i<n
    potencia ← potencia * x
    i ← i+1
finMientras
mostrar fin_linea, "La potencia es ", potencia
fin
```

Modifíquelo transformando la estructura **mientras** en una estructura para. Luego traduzca las dos versiones del programa a un lenguaje de programación como Java o C.

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int x, n;
6     int potencia = 1;
7
8     cout << "Introduzca la base y el exponente: ";
9     cin >> x >> n;
10
11    for (int i = 0; i < n; i++) {
12        potencia = potencia * x;
13    }
14
15    cout << "La potencia es " << potencia << endl;
16
17    return 0;
18
19 }
```



```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (9).exe
Introduzca la base y el exponente: 5
7
La potencia es 78125
-----
Process exited after 7.315 seconds with return value 0
Presione una tecla para continuar . . .
```

EJERCICIO 10:

10.- Desarrolle un programa que determine la paga bruta de cada uno de varios empleados. La compañía paga “tiempo normal” por las primeras 40 horas laboradas por cada empleado y paga “tiempo y medio” por todas las horas laboradas en exceso de 40 horas. Usted recibe una lista de los empleados de la compañía, el número de horas que cada empleado trabajó la semana anterior y el sueldo por hora de cada empleado. Su programa deberá introducir esta información para cada empleado y determinar y exhibir la paga bruta del empleado.

Por ejemplo (el texto en negrilla corresponde a los datos introducidos por el usuario):

```
Teclee horas laboradas (0 para terminar): 39
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 390
Teclee horas laboradas (0 para terminar): 40
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 400
Teclee horas laboradas (0 para terminar): 41
Teclee sueldo por hora del trabajador $ : 10
La paga es $: 415
```

```
1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     double horas, sueldoHora, pagaBruta;
6
7     cout << "Teclee horas laboradas (0 para terminar): ";
8     cin >> horas;
9
10    while (horas != 0) {
11        cout << "Teclee sueldo por hora del trabajador $: ";
12        cin >> sueldoHora;
13
14        if (horas <= 40) {
15            pagaBruta = horas * sueldoHora;
16        } else {
17            pagaBruta = (40 * sueldoHora) + ((horas - 40) * (1.5 * sueldoHora));
18        }
19
20        cout << "La paga es $: " << pagaBruta << endl << endl;
21
22        cout << "Teclee horas laboradas (0 para terminar): ";
23        cin >> horas;
24    }
25
26    cout << "Programa finalizado." << endl;
27
28    return 0;
29}
30
```

```
C:\Users\ADMIN\Desktop\Ejercicios de C++\pt 3 (10).exe
Teclee horas laboradas (0 para terminar): 5
Teclee sueldo por hora del trabajador $: 500
La paga es $: 2500

Teclee horas laboradas (0 para terminar): -
```