

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)
- Zapata Alanía Carolina (24140186)
- Avalos Palomino Ismael Jesús (24140158)

INFORME DE EJERCICIOS DE PROGRAMACIÓN EN JAVA

Elaborado por: Carolina Zapata

1. Realizar un algoritmo que muestre por pantalla la tabla de multiplicar del tres y del 9.

Algoritmo:

1. Mostrar el título "TABLA DEL 3" en pantalla.
2. Para cada número desde 0 hasta 12 (inclusive):
3. Calcular el producto de ese número por 3.
4. Mostrar en pantalla el resultado en el formato:
número x 3 = resultado
5. Mostrar separadores visuales en pantalla (-----).
6. Mostrar el título "TABLA DEL 9" en pantalla.
7. Para cada número desde 0 hasta 12 (inclusive):
8. Calcular el producto de ese número por 9.
9. Mostrar en pantalla el resultado en el formato:
a. número x 9 = resultado

Código fuente:

```

1  package Grupo1Problemas;
2  public class Primero {
3      //Realizar un algoritmo que muestre por pantalla la tabla de multiplicar del tres y del 9
4      public static void main(String[] args) {
5          System.out.println("TABLA DEL 3");
6          for(int i=0;i<=12;i+=1){
7              int producto = i*3;
8              System.out.println(i+" x 3 = " + producto);
9          }
10         System.out.println("-----");
11         System.out.println("-----");
12         System.out.println("-----");
13         System.out.println("TABLA DEL 9");
14         for(int i=0;i<=12;i++){
15             int producto = i*9;
16             System.out.println(i+" x 9 = " + producto);
17         }
18     }
19 }
```

Captura de resultado:

```

run:
TABLA DEL 3
0 × 3 = 0
1 × 3 = 3
2 × 3 = 6
3 × 3 = 9
4 × 3 = 12
5 × 3 = 15
6 × 3 = 18
7 × 3 = 21
8 × 3 = 24
9 × 3 = 27
10 × 3 = 30
11 × 3 = 33
12 × 3 = 36
-----
-----
TABLA DEL 9
0 × 9 = 0
1 × 9 = 9
2 × 9 = 18
3 × 9 = 27
4 × 9 = 36
5 × 9 = 45
6 × 9 = 54
7 × 9 = 63
8 × 9 = 72
9 × 9 = 81
10 × 9 = 90
11 × 9 = 99
12 × 9 = 108
BUILD SUCCESSFUL (total time: 0 seconds)

```

2. Dado un número entero, visualiza por pantalla la suma de todos los números pares desde 1 hasta ese número, la suma de todos los números impares desde 1 hasta ese número y la suma de todos los múltiplos de 5 desde 1 hasta ese número.

Algoritmo:

1. Pedir al usuario que ingrese un número entero.
2. Guardar ese número como "límite".
3. Inicializar tres variables para guardar sumas:
4. sumapares en 0.
5. sumaimpares en 0.
6. suma_multiplos_de_5 en 0.
7. Mostrar por pantalla el número ingresado.
8. Desde el número 0 hasta el número ingresado, hacer lo siguiente por cada número i:
9. Si i es par (es divisible entre 2), sumarlo a sumapares.
10. Si i es impar (no es divisible entre 2), sumarlo a sumaimpares.

11. Si i es divisible entre 5, sumarlo a `suma_multiplos_de_5` y mostrarlo por pantalla como múltiplo.
12. Mostrar el total de la suma de pares.
13. Mostrar el total de la suma de impares.
14. Mostrar el total de la suma de los múltiplos de 5.

Código fuente:

```

1  package Grupo1Problemas;
2  import java.util.Scanner;
3  public class Segundo {
4      /*Dado un número entero, visualiza por pantalla la suma de todos los números pares desde 1 hasta ese
5      número, la suma de todos los números impares desde 1 hasta ese número y la suma de todos los múltiplos
6      de 5 desde 1 hasta ese número*/
7      public static void main(String[] args) {
8          Scanner you=new Scanner (System.in);
9          System.out.print("Ingrese su numero: ");
10         int entero=you.nextInt();
11         int sumapares=0;
12         int sumaimpares=0;
13         int suma_multiplos_de_cinco=0;
14         System.out.println("El numero ingresado es: "+entero);
15         for(int i=0;i<=entero;i+=1){
16             if(i % 2 ==0){
17                 sumapares=i+sumapares;
18             }
19             if(i % 2 !=0){
20                 sumaimpares=i+sumaimpares;
21             }
22             if(i % 5 ==0){
23                 suma_multiplos_de_cinco=suma_multiplos_de_cinco+i;
24                 System.out.println("Los multiplos de 5 son : "+i);
25             }
26         }
27         System.out.println("La suma de los pares es : "+sumapares);
28         System.out.println("La suma de los impares es : "+sumaimpares);
29         System.out.println("La suma de los multiplos de 5 son : "+suma_multiplos_de_cinco);
30     }
31 }
```

Captura de resultado:

```

run:
Ingrese su numero: 12
El numero ingresado es: 12
Los multiplos de 5 son : 0
Los multiplos de 5 son : 5
Los multiplos de 5 son : 10
La suma de los pares es : 42
La suma de los impares es : 36
La suma de los multiplos de 5 son : 15
BUILD SUCCESSFUL (total time: 10 seconds)
```

3. Escriba un programa que muestre la suma de los 20 primeros múltiplos de 3

Algoritmo:

1. Inicializar una variable llamada suma con valor 0.
2. Mostrar en pantalla el mensaje: "20 primeros múltiplos de 3".
3. Para cada número desde 1 hasta 20:
4. Calcular el múltiplo de 3 multiplicando el número por 3.
5. Mostrar el múltiplo en pantalla.
6. Sumar el múltiplo a la variable suma.
7. Al terminar el ciclo, mostrar en pantalla la suma total de los 20 primeros múltiplos de 3

Código fuente:

```

1 package Grupo1Problemas;
2
3 public class Tercero {
4
5     //Escriba un programa que muestre la suma de los 20 primeros múltiplos de 3
6
7     public static void main(String[] args) {
8
9         System.out.println("20 primeros multiplos de 3");
10        int suma = 0; //variable que almacenará la suma de multiplos
11
12        for(int i=1;i<=20;i+=1){
13            int multiplos = i*3;
14            System.out.print(multiplos + " ");
15
16            suma += multiplos;
17        }
18        System.out.println("\nSuma de los 20 primeros multiplos de 3: " + suma);
19    }
20
21
22 }
```

Captura de resultado:

```

run:
20 primeros multiplos de 3
3 6 9 12 15 18 21 24 27 30 33 36 39 42 45 48 51 54 57 60
Suma de los 20 primeros multiplos de 3: 630
BUILD SUCCESSFUL (total time: 0 seconds)
```

4. Escriba un programa que: rota, hacia la izquierda, los dígitos de un número tantas veces como indique el promedio de sus dígitos. Ej. si el valor de entrada es 123 se obtendrá el número 312.

Algoritmo:

1. Pedir al usuario que ingrese un número entero.
2. Leer el número como una cadena (string) para poder manipular sus dígitos.
3. Inicializar una variable llamada suma con valor 0.
4. Recorrer cada carácter del número, uno por uno:
 - Convertir el carácter a su valor numérico.
 - Sumar el valor numérico al total de suma.
5. Calcular el promedio de los dígitos realizando una división entera entre suma y la cantidad de dígitos del número.
6. Mostrar el promedio de los dígitos al usuario.
7. Realizar una rotación a la izquierda del número tantas veces como indique el promedio:

Para cada rotación, mover el primer dígito del número al final de la cadena.
8. Mostrar el número después de haber sido rotado.

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class Cuarto {
6
7     /*Escriba un programa que: rota, hacia la izquierda, los dígitos de un número tantas veces como indique
8     el promedio de sus dígitos. Ej. si el valor de entrada es 123 se obtendrá el número 312.*/
9
10    public static void main(String[] args) {
11        Scanner sc = new Scanner(System.in);
12        System.out.print("Ingresa un numero entero: ");
13        String numero = sc.nextLine(); // Usamos String para manipular los dígitos
14
15        int suma = 0;
16        for (int i = 0; i < numero.length(); i+=1) {
17            suma += Character.getNumericValue(numero.charAt(i));
18        }
19
20        int promedio = suma / numero.length(); // División entera
21        System.out.println("Promedio de los digitos: " + promedio);
22
23        // Rotar a la izquierda tantas veces como el promedio
24        for (int i = 0; i < promedio; i++) {
25            numero = numero.substring(1) + numero.charAt(0); // Corta el primer dígito y lo pone al final
26        }
27
28        System.out.println("Resultado despues de rotar: " + numero);
29    }
30}

```

Captura de resultado:

```

run:
Ingresa un numero entero: 3685
Promedio de los digitos: 5
Resultado despues de rotar: 6853
BUILD SUCCESSFUL (total time: 4 seconds)

```

5. Escriba un programa que lea de la entrada estándar valores que representan años e indique si son o no años bisiestos. El programa seguirá leyendo años hasta un máximo de 10 o hasta que haya leído 3 años bisiestos.

Algoritmo:

1. Inicializar la variable totalAnios en 0 (para contar los años ingresados).
2. Inicializar la variable bisiestos en 0 (para contar los años bisiestos encontrados).
3. Iniciar un ciclo mientras totalAnios sea menor a 10 y bisiestos sea menor a 3:
 - Pedir al usuario que ingrese un año.
 - Leer el valor del año.
 - Incrementar totalAnios en 1.
 - Verificar si el año es bisiesto:

- Si el año es divisible entre 4, pero no entre 100, o si es divisible entre 400, entonces es un año bisiesto.
 - Si es bisiesto, imprimir que el año es bisiesto y aumentar el contador bisiesto en 1.
 - Si no es bisiesto, imprimir que el año no es bisiesto.
4. Cuando se termine el ciclo, mostrar el total de años ingresados y cuántos años bisiestos se han encontrado.
5. Terminar el programa.

Código fuente:

```

1  package Grupo1Problemas;
2
3  import java.util.Scanner;
4
5  public class Quinto {
6
7      /*Escriba un programa que lea de la entrada estándar valores que representan años e indique si son o no
8      años bisiestos. El programa seguirá leyendo años hasta un máximo de 10 o hasta que haya leído 3 años
9      bisiestos.*/
10
11     public static void main(String[] args) {
12
13         Scanner scanner = new Scanner(System.in);
14
15         int totalAnios = 0;
16         int bisiestos = 0;
17
18         while (totalAnios < 10 && bisiestos < 3) {
19             System.out.print("Ingresa un año: ");
20             int anio = scanner.nextInt();
21
22             totalAnios++;
23
24             if ((anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0)) {
25                 System.out.println(anio + " es un año bisiesto.");
26                 bisiestos++;
27             } else {
28                 System.out.println(anio + " no es un año bisiesto.");
29             }
30
31         }
32
33         System.out.println("\nPrograma finalizado. Total de años ingresados: " + totalAnios);
34         System.out.println("Años bisiestos encontrados: " + bisiestos);
35     }
}

```

Captura de resultado:

```

run:
Ingresa un anio: 2050
2050 no es un anio bisiesto.
Ingresa un anio: 1400
1400 no es un anio bisiesto.
Ingresa un anio: 1560
1560 es un anio bisiesto.
Ingresa un anio: 2025
2025 no es un anio bisiesto.
Ingresa un anio: 1160
1160 es un anio bisiesto.
Ingresa un anio: 1340
1340 es un anio bisiesto.

Programa finalizado. Total de anios ingresados: 6
Anios bisiestos encontrados: 3
BUILD SUCCESSFUL (total time: 50 seconds)

```

6. Escriba un programa que calcule de forma aproximada el seno de un ángulo comprendido entre el valor del ángulo se leerá desde la entrada estándar, y se comprobará que se encuentra entre los límites fijados.

Algoritmo:

1. Solicitar al usuario que ingrese un valor de ángulo en grados (entre 0 y 360).
2. Leer el valor del ángulo.
3. Comprobar si el valor del ángulo está dentro del rango permitido (entre 0 y 360 grados):
 - Si el valor del ángulo es menor que 0 o mayor que 360, mostrar un mensaje indicando que el ángulo está fuera de rango y terminar el programa.
4. Si el valor está dentro del rango, proceder con los siguientes pasos:
 - Convertir el valor del ángulo de grados a radianes usando la fórmula: radianes = grados * $\pi / 180$.
 - Calcular el seno del ángulo usando una aproximación de la serie de Taylor con 5 términos:
 - Para cada término de la serie, calcular el exponente ($2 * i + 1$) y el signo alternado utilizando `Math.pow(-1, i)`.
 - Calcular el factorial del exponente y usarlo para calcular el término de la serie: término = $(Math.pow(-1, i) * Math.pow(x, exponente)) / factorial(exponente)$.

- Sumar los términos para obtener la aproximación del seno.
5. Mostrar el valor aproximado del seno calculado.
 6. Mostrar el valor real del seno utilizando Math.sin(radianes) para comparar con la aproximación.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class Sexto {
4     /*Escriba un programa que calcule de forma aproximada el seno de un ángulo comprendido entre
5     El valor del ángulo se leerá desde la entrada estándar, y se comprobará que se encuentra entre los límites
6     fijados.*/
7     // Método para convertir grados a radianes
8     public static double gradosARadianes(double grados) {
9         return grados * Math.PI / 180;
10    }
11    // Método para calcular el factorial
12    public static long factorial(int n) {
13        long resultado = 1;
14        for (int i = 2; i <= n; i++) {
15            resultado *= i;
16        }
17        return resultado;
18    }
19    // Método para calcular el seno con 5 términos de la serie de Taylor
20    public static double calcularSeno(double x) {
21        double seno = 0;
22        for (int i = 0; i < 5; i++) {
23            int exponente = 2 * i + 1;
24            double termino = Math.pow(-1, i) * Math.pow(x, exponente) / factorial(exponente);
25            seno += termino;
26        }
27        return seno;
28    }
29    // Método principal
30    public static void main(String[] args) {
31        Scanner scanner = new Scanner(System.in);
32        double angulo;
33        System.out.print("Ingresa un angulo en grados (entre 0 y 360): ");
34        angulo = scanner.nextDouble();
35        if (angulo < 0 || angulo > 360) {
36            System.out.println("Angulo fuera de rango permitido. Intenta con un valor entre 0° y 360°.");
37        } else {
38            double radianes = gradosARadianes(angulo);
39            double senoAprox = calcularSeno(radianes);
40            System.out.println("Aproximacion del seno de " + angulo + ": " + senoAprox);
41            System.out.println("Valor real con Math.sin: " + Math.sin(radianes));
42        }
43    }
44}

```

Captura de resultado:

```

run:
Ingresa un angulo en grados (entre 0 y 360): 250
Aproximacion del seno de 250.0: -0.6968137969996808
Valor real con Math.sin: -0.9396926207859082
BUILD SUCCESSFUL (total time: 7 seconds)

```

7. *El siguiente programa genera un número aleatorio comprendido entre 1 y el valor de la constante MAXIMO y posteriormente reta al usuario a que lo adivine. Para ello utiliza la función random que genera un número aleatorio x donde $0 \leq x < 1$.]

Algoritmo:

1. Definir una constante MAXIMO con valor 100.
2. Generar un número aleatorio numeroSecreto entre 1 y MAXIMO.
3. Inicializar la variable acierto como falso.
4. Mientras acierto sea falso, repetir:
 - o Mostrar mensaje solicitando un número entre 1 y MAXIMO.
 - o Leer el número ingresado por el usuario.
 - o Si el número ingresado está fuera del rango [1, MAXIMO]:
 - Mostrar mensaje de error por número fuera de rango.
 - Continuar con la siguiente iteración del bucle.
 - o Comparar el número ingresado con numeroSecreto:
 - Si son iguales:
 - Mostrar mensaje de acierto.
 - Cambiar acierto a verdadero para salir del bucle.
 - Si el número ingresado es menor:
 - Mostrar mensaje indicando que el número es mayor.
 - Si el número ingresado es mayor:
 - Mostrar mensaje indicando que el número es menor.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class Septimo {
4
5     /*El siguiente programa genera un número aleatorio comprendido entre 1 y el valor de la constante
6     MAXIMO y posteriormente reta al usuario a que lo adivine. Para ello utiliza la función random que
7     genera un número aleatorio x donde 0 ≤ x < 1.*/
8
9     public static void main(String[] args) {
10         final int MAXIMO = 100;
11         int numeroSecreto = (int) (Math.random() * MAXIMO) + 1;
12         boolean acierto = false;
13         Scanner scanner = new Scanner(System.in);
14
15         while (!acierto) {
16             System.out.println("\nIntroduzca un numero entre 1 y " + MAXIMO + ":");
17             int n = scanner.nextInt();
18
19             // Validación: ¿está dentro del rango permitido?
20             if (n < 1 || n > MAXIMO) {
21                 System.out.println("Número fuera de rango. Intenta con un numero entre 1 y " + MAXIMO + ".");
22                 continue; // salta a la siguiente iteración sin comparar
23             }
24
25             if (n == numeroSecreto) {
26                 System.out.println("¡¡Acertaste!!!");
27                 acierto = true;
28             } else if (n < numeroSecreto) {
29                 System.out.println("El numero es mayor.");
30             } else {
31                 System.out.println("El numero es menor.");
32             }
33         }
34     }
35 }
```

Captura de resultado:

```

run:

Introduzca un numero entre 1 y 100:
60
El numero es menor.

Introduzca un numero entre 1 y 100:
30
El numero es mayor.

Introduzca un numero entre 1 y 100:
50
El numero es menor.

Introduzca un numero entre 1 y 100:
40
El numero es menor.

Introduzca un numero entre 1 y 100:
33
El numero es mayor.

Introduzca un numero entre 1 y 100:
37
El numero es mayor.

Introduzca un numero entre 1 y 100:
39
El numero es menor.

Introduzca un numero entre 1 y 100:
38
♦♦Acertaste!!
BUILD SUCCESSFUL (total time: 28 seconds)
```

8. Escriba un programa que imprima la tabla de multiplicar de un número que se leerá de la entrada estándar.

Algoritmo:

1. Solicitar al usuario que ingrese un número.
2. Leer el número ingresado y guardarlo en la variable numero.
3. Mostrar un encabezado indicando que se imprimirá la tabla del número.
4. Para i desde 1 hasta 12, hacer:
 - Calcular el producto numero × i.
 - Imprimir el resultado en formato: numero × i = resultado.

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class Octavo {
6
7     /*Escriba un programa que imprima la tabla de multiplicar de un número que se leerá de la entrada
8     estándar.*/
9
10    public static void main(String[] args) {
11
12        Scanner scanner = new Scanner(System.in);
13
14        System.out.print("Introduzca un numero: ");
15        int numero = scanner.nextInt();
16
17        System.out.println("La tabla de multiplicar del " + numero + " es:\n");
18
19        for (int i = 1; i <= 12; i++) {
20            System.out.printf("%2d x %2d = %3d\n", numero, i, numero * i);
21        }
22    }
23 }
```

Captura de resultado:

```

run:
Introduzca un numero: 8
La tabla de multiplicar del 8 es:

 8 x  1 =    8
 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 =   80
 8 x 11 =   88
 8 x 12 =   96

BUILD SUCCESSFUL (total time: 10 seconds)

```

9. Escriba un programa que muestre la serie de Fibonacci para los valores comprendidos entre n y m.

Algoritmo:

1. Solicitar al usuario que introduzca el valor inicial n.
2. Leer el valor n.
3. Solicitar al usuario que introduzca el valor final m.
4. Leer el valor m.
5. Si $n > m$, mostrar un mensaje de error y finalizar el programa.
6. Mostrar el mensaje: "Serie de Fibonacci entre n y m:".
7. Inicializar $a = 0$ y $b = 1$ (primeros términos de Fibonacci).
8. Mientras $a \leq m$, repetir:
 - Si $a \geq n$, imprimir el valor de a.
 - Calcular el siguiente término: $\text{temp} = a + b$.
 - Actualizar $a = b$.
 - Actualizar $b = \text{temp}$.
9. Imprimir salto de línea al final para mejor presentación.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class Noveno {
4     //Escriba un programa que muestre la serie de Fibonacci para los valores comprendidos entre n y m.
5     public static void main(String[] args) {
6         Scanner scanner = new Scanner(System.in);
7
8         System.out.print("Introduce el valor inicial (n): ");
9         int n = scanner.nextInt();
10
11        System.out.print("Introduce el valor final (m): ");
12        int m = scanner.nextInt();
13
14        if (n > m) {
15            System.out.println("El valor inicial (n) debe ser menor o igual que el valor final (m).");
16            return;
17        }
18
19        System.out.println("Serie de Fibonacci entre " + n + " y " + m + ":");
20
21        int a = 0, b = 1;
22        while (a <= m) {
23            if (a >= n) {
24                System.out.print(a + " ");
25            }
26            int temp = a + b;
27            a = b;
28            b = temp;
29        }
30
31        System.out.println(); // Salto de línea al final
32    }
33}

```

Captura de resultado:

```

run:
Introduce el valor inicial (n): 0
Introduce el valor final (m): 13
Serie de Fibonacci entre 0 y 13:
0 1 1 2 3 5 8 13
BUILD SUCCESSFUL (total time: 3 seconds)

```

10. Traduzca un número binario introducido desde el teclado, al sistema hexadecimal (base 16) de forma directa, es decir sin convertir el número binario al sistema decimal.

Algoritmo:

1. Solicitar al usuario que introduzca un número binario (como cadena de texto).
2. Leer el número binario (binario).
3. Calcular la longitud del número binario.
4. Calcular cuántos ceros faltan para que la longitud sea múltiplo de 4:
 - cerosNecesarios = $(4 - (\text{longitud \% 4})) \% 4$.

5. Crear una nueva cadena (binarioCompleto) y rellenarla con ceros a la izquierda hasta que tenga longitud múltiplo de 4.
6. Añadir el número binario original al final de binarioCompleto.
7. Inicializar una cadena vacía hexadecimal.
8. Recorrer binarioCompleto de 4 en 4 caracteres:
 - Obtener un grupo de 4 bits.
 - Evaluar el grupo usando un switch y agregar el dígito hexadecimal correspondiente a hexadecimal.
 - Si el grupo no coincide con ningún patrón de 4 bits válido, mostrar un mensaje de error y terminar el programa.
9. Mostrar el resultado final: "Representación hexadecimal: " + hexadecimal.

Código fuente:

```

1 package GrupolProblemas;
2 import java.util.Scanner;
3 public class Decimo {
4     /*Traduzca un número binario introducido desde el teclado, al sistema hexadecimal (base 16) de forma
5 directa, es decir sin convertir el número binario al sistema decimal.*/
6     public static void main(String[] args) {
7         Scanner scanner = new Scanner(System.in);
8         System.out.print("Introduce un numero binario: ");
9         String binario = scanner.nextLine();
10        // Asegurar que el número tenga un número de bits múltiplo de 4 (rellenar con ceros a la izquierda)
11        int longitud = binario.length();
12        int cerosNecesarios = (4 - (longitud % 4)) % 4;
13        StringBuilder binarioCompleto = new StringBuilder();
14        for (int i = 0; i < cerosNecesarios; i++) {
15            binarioCompleto.append("0");
16        }
17        binarioCompleto.append(binario);
18        // Mapeo directo binario → hexadecimal
19        StringBuilder hexadecimal = new StringBuilder();
20        for (int i = 0; i < binarioCompleto.length(); i += 4) {
21            String grupo4 = binarioCompleto.substring(i, i + 4);
22            switch (grupo4) {
23                case "0000": hexadecimal.append("0"); break;
24                case "0001": hexadecimal.append("1"); break;
25                case "0010": hexadecimal.append("2"); break;
26                case "0011": hexadecimal.append("3"); break;
27                case "0100": hexadecimal.append("4"); break;
28                case "0101": hexadecimal.append("5"); break;
29                case "0110": hexadecimal.append("6"); break;
30                case "0111": hexadecimal.append("7"); break;
31                case "1000": hexadecimal.append("8"); break;
32                case "1001": hexadecimal.append("9"); break;
33                case "1010": hexadecimal.append("A"); break;
34                case "1011": hexadecimal.append("B"); break;
35                case "1100": hexadecimal.append("C"); break;
36                case "1101": hexadecimal.append("D"); break;
37                case "1110": hexadecimal.append("E"); break;
38                case "1111": hexadecimal.append("F"); break;
39                default:
40                    System.out.println("Entrada invalida. Asegurate de introducir solo 0s y 1s.");
41                    return;
42            }
43        }
44        System.out.println("Representacion hexadecimal: " + hexadecimal);
45    }
46}

```

Captura de resultado:

```
run:
Introduce un numero binario: 101010100001
Representacion hexadecimal: AA1
BUILD SUCCESSFUL (total time: 9 seconds)
```

11. Una empresa fabrica dos productos A y B y desea saber cuál de ellos es el más aceptado en el mercado. Para ello realiza una encuesta y, por cada persona interrogada se obtiene un par de valores: el primer valor del par indica la aceptación o no del producto A según sea 1 o 0; el segundo valor del par corresponde al producto B. Por ejemplo, el par (1,0) significa que el encuestado acepta el producto A pero no el B. Se pide:

- Total de encuestados
- Porcentaje de consumidores que aceptan:
- El producto A
- El producto B
- Los dos productos
- El producto A pero no el B
- El producto B pero no el A
- Ninguno de los dos productos.

Algoritmo:

1. Inicializar contadores:
 - aceptarA, aceptarB, ambos, soloA, soloB, ninguno, total $\leftarrow 0$.
2. Mostrar instrucciones al usuario:
 - Que ingrese pares (A, B) con valores 0 o 1.
 - Finalizar la entrada con -1 -1.
3. Repetir hasta que se ingrese -1 -1:
 - Pedir el par (A, B).
 - Validar que A y B sean 0 o 1.
 - Si no lo son, mostrar error y continuar.
 - Incrementar total.
 - Contabilizar:

- Si A = 1 → aceptarA++
- Si B = 1 → aceptarB++
- Si A = 1 y B = 1 → ambos++
- Si A = 1 y B = 0 → soloA++
- Si A = 0 y B = 1 → soloB++
- Si A = 0 y B = 0 → ninguno++

4. Si no se ingresó ningún dato válido (total = 0):

- Mostrar mensaje de error y terminar.

5. Mostrar resultados:

- Total de encuestados.
- Porcentajes:
 - Producto A → aceptarA × 100 ÷ total
 - Producto B → aceptarB × 100 ÷ total
 - Ambos productos → ambos × 100 ÷ total
 - Solo A → soloA × 100 ÷ total
 - Solo B → soloB × 100 ÷ total
 - Ninguno → ninguno × 100 ÷ total

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class Onceavo {
6     /*Una empresa fabrica dos productos A y B y desea saber cuál de ellos es el más aceptado en el
7 mercado. Para ello realiza una encuesta y, por cada persona interrogada se obtiene un par de valores: el
8 primer valor del par indica la aceptación o no del producto A según sea 1 o 0; el segundo valor del par
9 corresponde al producto B. Por ejemplo, el par (1,0) significa que el encuestado acepta el producto A pero
10 no el B. Se pide:
11 - Total de encuestados
12 - Porcentaje de consumidores que aceptan:
13 - El producto A
14 - El producto B
15 - Los dos productos
16 - El producto A pero no el B
17 - El producto B pero no el A
18 - Ninguno de los dos productos. */
19
20 public static void main(String[] args) {
21
22     Scanner scanner = new Scanner(System.in);
23
24     int aceptarA = 0;
25     int aceptarB = 0;
26     int ambos = 0;
27     int soloA = 0;
28     int soloB = 0;
29     int ninguno = 0;
30     int total = 0;
31
32     System.out.println("Introduce las respuestas de los encuestados en formato (A,B).");
33     System.out.println("Usa 1 para 'si' y 0 para 'no'. Ejemplo: 1 0");
34     System.out.println("Escribe -1 -1 para terminar la encuesta.");
35
36     while (true) {
37         System.out.print("Respuesta " + (total + 1) + ": ");
38         int a = scanner.nextInt();
39         int b = scanner.nextInt();
40         if (a == -1 & b == -1) {
41             break;
42         }
43         if ((a != 0 && a != 1) || (b != 0 && b != 1)) {
44             System.out.println("Entrada invalida. Solo se permiten 0 o 1.");
45             continue;
46         }
47
48         total++;
49         if (a == 1) aceptarA++;
50         if (b == 1) aceptarB++;
51         if (a == 1 && b == 1) ambos++;
52         if (a == 1 && b == 0) soloA++;
53         if (a == 0 && b == 1) soloB++;
54         if (a == 0 && b == 0) ninguno++;
55     }
56
57     if (total == 0) {
58         System.out.println("No se ingresaron datos validos.");
59         return;
60     }
61     System.out.println("\nResultados de la encuesta:");
62     System.out.println("Total de encuestados: " + total);
63     System.out.printf("%% que aceptan el producto A: %.2f%%\n", (aceptarA * 100.0 / total));
64     System.out.printf("%% que aceptan el producto B: %.2f%%\n", (aceptarB * 100.0 / total));
65     System.out.printf("%% que aceptan ambos productos: %.2f%%\n", (ambos * 100.0 / total));
66     System.out.printf("%% que aceptan solo el producto A: %.2f%%\n", (soloA * 100.0 / total));
67     System.out.printf("%% que aceptan solo el producto B: %.2f%%\n", (soloB * 100.0 / total));
68     System.out.printf("%% que no aceptan ninguno: %.2f%%\n", (ninguno * 100.0 / total));
69 }

```

Captura de resultado:

```

run:
Introduce las respuestas de los encuestados en formato (A,B).
Usa 1 para 'si' y 0 para 'no'. Ejemplo: 1 0
Escribe -1 -1 para terminar la encuesta.
Respuesta 1: 0 1
Respuesta 2: 1 0
Respuesta 3: 1 0
Respuesta 4: 1 0
Respuesta 5: 1 1
Respuesta 6: -1 -1

Resultados de la encuesta:
Total de encuestados: 5
% que aceptan el producto A: 80,00%
% que aceptan el producto B: 40,00%
% que aceptan ambos productos: 20,00%
% que aceptan solo el producto A: 60,00%
% que aceptan solo el producto B: 20,00%
% que no aceptan ninguno: 0,00%
BUILD SUCCESSFUL (total time: 21 seconds)

```

12. Diseñe un diagrama que acepte un número escrito del 1 al 20 en cifras arábigas y lo visualice en números romanos.

Algoritmo:

1. Mostrar mensaje al usuario:
→ “Ingrese un número entre 1 y 20”.
2. Leer el número ingresado (numero).
3. Verificar si el número está en el rango:
 - Si numero < 1 o numero > 20:
 - Mostrar mensaje de error:
→ “Número fuera de rango”.
4. Si está en el rango permitido:
 - Llamar a la función convertirARomano(numero).
 - Mostrar el resultado:
→ “El número X en romano es: Y”.
5. Fin.

Código fuente:

```

1  package Grupo1Problemas;
2  import java.util.Scanner;
3  public class Doceavo {
4      /*Diseñe un diagrama que acepte un número escrito del 1 al 20 en cifras arábigas y lo
5      visualice en números romanos.*/
6      public static void main(String[] args) {
7          // Crear un scanner para obtener la entrada del usuario
8          Scanner scanner = new Scanner(System.in);
9
10         // Pedir al usuario que ingrese un número entre 1 y 20
11         System.out.print("Ingrese un numero entre 1 y 20: ");
12         int numero = scanner.nextInt();
13
14         // Verificar si el número está en el rango permitido
15         if (numero < 1 || numero > 20) {
16             System.out.println("Número fuera de rango. Por favor ingrese un numero entre 1 y 20.");
17         } else {
18             // Llamar a la función que convierte el número a romano y mostrar el resultado
19             System.out.println("El numero " + numero + " en romano es: " + convertirARomano(numero));
20         }
21         // Cerrar el scanner
22         scanner.close();
23     }
24
25     // Función que convierte el número a romano
26     public static String convertirARomano(int numero) {
27         // Arreglo con los números romanos del 1 al 20
28         String[] numerosRomanos = {
29             "", "I", "II", "III", "IV", "V", "VI", "VII", "VIII", "IX",
30             "X", "XI", "XII", "XIII", "XIV", "XV", "XVI", "XVII", "XVIII", "XIX", "XX"
31         };
32         // Retornar el número romano correspondiente
33         return numerosRomanos[numero];
34     }

```

Captura de resultado:

```

run:
Ingrese un numero entre 1 y 20: 15
El numero 15 en romano es: XV

```

13.

Algoritmo:

1. suma empieza en 0.
2. producto empieza en 1 como BigInteger.
3. Bucle for ($i = 20; i \leq 400; i++$):
 - o Si $i \% 2 == 0$ (es par):
 - Se suma a suma.
 - Se multiplica con producto usando BigInteger.valueOf(i).
4. Al final:
 - o Se imprime la suma total.
 - o Se imprime el producto total.

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.math.BigInteger;
4
5 public class DecimoTercero {
6
7     /*Calcular y visualizar la suma y el producto de los números pares comprendidos entre 20 y
8     400 ambos inclusive/
9
10    public static void main(String[] args) {
11        // Inicializar las variables para la suma y el producto (BigInteger para el producto)
12        int suma = 0;
13        BigInteger producto = BigInteger.ONE; // Usamos BigInteger para manejar números muy grandes
14
15        // Recorrer los números del 20 al 400
16        for (int i = 20; i <= 400; i++) {
17            if (i % 2 == 0) { // Verificar si el número es par
18                suma += i; // Sumar el número par
19                // Multiplicar el número par usando BigInteger
20                producto = producto.multiply(BigInteger.valueOf(i));
21            }
22        }
23
24        // Mostrar los resultados
25        System.out.println("La suma de los numeros pares entre 20 y 400 es: " + suma);
26        System.out.println("El producto de los numeros pares entre 20 y 400 es: " + producto);
27    }
28 }
```

Captura de resultado:

```

run:
La suma de los numeros pares entre 20 y 400 es: 40110
El producto de los numeros pares entre 20 y 400 es: 682110569315184795844051020839627918763336724759301153536786354782855085354604085
BUILD SUCCESSFUL (total time: 0 seconds)
```

14. Escribir un algoritmo que lea una secuencia de numeros positivos acabada en 0 y encuentre el menor, el mayor y el promedio de los numeros leidos.

Algoritmo:

Inicio

1. Mostrar mensaje:
→ “Ingresa una secuencia de números positivos (termina con 0):”.
2. Inicializar:
 - menor = ∞ → usando Integer.MAX_VALUE.
 - mayor = $-\infty$ → usando Integer.MIN_VALUE.
 - suma = 0
 - cantidad = 0
3. Mientras el número leído $\neq 0$:
 - o Si el número < 0 :

- Mostrar mensaje de error:
→ “Por favor ingresa solo números positivos”.
- Si el número > 0 :
 - Actualizar menor si el número es más pequeño.
 - Actualizar mayor si el número es más grande.
 - Sumarlo a suma.
 - Incrementar cantidad.
- 4. Si se ingresaron números válidos (cantidad > 0):
 - Calcular el promedio:
→ $\text{promedio} = \text{suma} \div \text{cantidad}$.
 - Mostrar:
 - “El número menor es: ...”
 - “El número mayor es: ...”
 - “El promedio de los números es: ...”
- 5. Si no se ingresaron números:
 - Mostrar:
→ “No se ingresaron números positivos”.

Fin

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class DecimoCuarto {
4     /*Escribir un algoritmo que lea una secuencia de numeros positivos acabada en 0 y encuentre el menor,
5     el mayor y el promedio de los numeros leidos. */
6     public static void main(String[] args) {
7         // Crear un scanner para leer los números ingresados por el usuario
8         Scanner scanner = new Scanner(System.in);
9         // Inicializar las variables
10        int numero, menor = Integer.MAX_VALUE, mayor = Integer.MIN_VALUE;
11        int suma = 0, cantidad = 0;
12        // Leer números hasta que el usuario ingrese 0
13        System.out.println("Ingresa una secuencia de numeros positivos (termina con 0):");
14        // Bucle para leer los números
15        while (true) {
16            numero = scanner.nextInt();
17            // Si el número es 0, salir del bucle
18            if (numero == 0) {
19                break;
20            }
21            // Validar que el número sea positivo
22            if (numero > 0) {
23                // Actualizar el menor y el mayor número
24                if (numero < menor) {
25                    menor = numero;
26                }
27                if (numero > mayor) {
28                    mayor = numero;
29                }
30                // Sumar el número y aumentar la cantidad de números leídos
31                suma += numero;
32                cantidad++;
33            } else {
34                System.out.println("Por favor ingresa solo numeros positivos.");
35            }
36        }
37        // Verificar si se ingresaron números positivos
38        if (cantidad > 0) {
39            // Calcular el promedio
40            double promedio = (double) suma / cantidad;
41            // Mostrar los resultados
42            System.out.println("El numero menor es: " + menor);
43            System.out.println("El numero mayor es: " + mayor);
44            System.out.println("El promedio de los numeros es: " + promedio);
45        } else {
46            System.out.println("No se ingresaron numeros positivos.");
47        }
48        // Cerrar el scanner
49        scanner.close();
50    }
51}

```

Captura de resultado:

```

run:
Ingresa una secuencia de numeros positivos (termina con 0):
8 8 5 6 3 9 2 54 5 8 4 3 6 2 1 5 8 7 2 3 99 8 5 54 42 3 565 0
El numero menor es: 1
El numero mayor es: 565
El promedio de los numeros es: 34.25925925925926
BUILD SUCCESSFUL (total time: 24 seconds)

```

15. Escribir un algoritmo que calcule y escriba los N números primeros términos de la sucesión de Fibonacci (siendo N un numero natural leído). la sucesión de números enteros de fibonacci se define de la forma siguiente

$f_1 = f_2 = f_1$

$f_n = f_{n-2} + f_{n-1}$ con $n > 2$

los primeros términos de esta sucesión son:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, ...

1. Algoritmo:

Inicio

1. Mostrar el mensaje:
"Ingrese el número de términos de la sucesión de Fibonacci:"
2. Leer el valor de N.
3. Si $N \leq 0$ entonces:
 - Mostrar: "Por favor ingrese un número natural mayor que 0."
 - Finalizar el algoritmo.
4. Si $N \geq 1$, entonces:
 - Mostrar 1
5. Si $N \geq 2$, entonces:
 - Mostrar otro 1
6. Inicializar dos variables:
 - $a \leftarrow 1$ (término anterior)
 - $b \leftarrow 1$ (término actual)
7. Para i desde 3 hasta N hacer:
 - $c \leftarrow a + b$ (siguiente término)
 - Mostrar c
 - $a \leftarrow b$
 - $b \leftarrow c$

Fin del algoritmo.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class DecimoQuinto {
4     /*Escribir un algoritmo que calcule y escriba los N números primeros términos de la sucesión de
5 fibonacci(siendo N un numero natural leído). la sucesión de números enteros de fibonacci se define de la
6 forma siguiente
7 f1=f2=1
8 fn = fn - 2 + fn-1 con n>2
9 los primeros términos de esta sucesión son:
10 1,1,2,3,5,8,13,21,34,55,89,144,233,... */
11 public static void main(String[] args) {
12     // Crear un scanner para obtener la entrada del usuario
13     Scanner scanner = new Scanner(System.in);
14     // Pedir al usuario que ingrese el número N
15     System.out.print("Ingrese el numero de terminos de la sucesion de Fibonacci: ");
16     int N = scanner.nextInt();
17     // Validar que N sea un número natural positivo
18     if (N <= 0) {
19         System.out.println("Por favor ingrese un numero natural mayor que 0.");
20     } else {
21         // Llamar a la función para calcular y mostrar los primeros N términos de Fibonacci
22         System.out.println("Los primeros " + N + " terminos de la sucesion de Fibonacci son:");
23         mostrarFibonacci(N);
24     }
25     // Cerrar el scanner
26     scanner.close();
27 }
28 // Función que calcula y muestra los N primeros términos de la sucesión de Fibonacci
29 public static void mostrarFibonacci(int N) {
30     // Variables para los dos primeros términos de Fibonacci
31     int a = 1, b = 1;
32     // Mostrar los primeros dos términos
33     if (N >= 1) System.out.print(a + " ");
34     if (N >= 2) System.out.print(b + " ");
35     // Calcular y mostrar los siguientes términos
36     for (int i = 3; i <= N; i++) {
37         int c = a + b; // Calcular el siguiente término
38         System.out.print(c + " "); // Mostrar el término
39         a = b; // Actualizar a para el siguiente término
40         b = c; // Actualizar b para el siguiente término
41     }
42     System.out.println(); // Salto de línea al final
43 }
44 }
```

Captura de resultado:

```

run:
Ingrese el numero de terminos de la sucesion de Fibonacci: 12
Los primeros 12 terminos de la sucesion de Fibonacci son:
1 1 2 3 5 8 13 21 34 55 89 144
BUILD SUCCESSFUL (total time: 8 seconds)
```

16. Un numero natural se considera perfecto cuando es igual a la suma de todos sus divisores , primos o no, excepto el mismo. por ejemplo; $28=1+2+4+7+14$ escribir si un numero natural positivo es perfecto si o no?

Algoritmo:

Inicio

1. Mostrar el mensaje:
"Ingrese un número natural positivo:"
2. Leer el valor de numero
3. Si numero ≤ 0 entonces:
 - Mostrar: "Por favor ingrese un número natural positivo."
 - Finalizar el algoritmo.
4. Inicializar sumaDivisores $\leftarrow 0$
5. Para i desde 1 hasta numero - 1 hacer:
 - Si numero % i = 0 entonces:
 - sumaDivisores \leftarrow sumaDivisores + i
6. Si sumaDivisores = numero entonces:
 - Mostrar: "El número [numero] es perfecto."
 - Sino:
 - Mostrar: "El número [numero] no es perfecto."

Fin

Código fuente:

```

1  package Grupo1Problemas;
2  import java.util.Scanner;
3  public class DecimoSexto {
4      /*Un numero natural se considera perfecto cuando es igual a la suma de todos sus divisores , primos o
5      no, excepto el mismo. por ejemplo; 28=1+2+4+7+14 escribir si un numero natural positivo es perfecto si
6      o no*/
7  public static void main(String[] args) {
8      // Crear un scanner para leer la entrada del usuario
9      Scanner scanner = new Scanner(System.in);
10     // Pedir al usuario que ingrese un número positivo
11     System.out.print("Ingrese un numero natural positivo: ");
12     int numero = scanner.nextInt();
13     // Validar que el número sea positivo
14     if (numero <= 0) {
15         System.out.println("Por favor ingrese un numero natural positivo.");
16     } else {
17         // Llamar a la función para verificar si el número es perfecto
18         if (esPerfecto(numero)) {
19             System.out.println("El numero " + numero + " es perfecto.");
20         } else {
21             System.out.println("El numero " + numero + " no es perfecto.");
22         }
23     }
24     // Cerrar el scanner
25     scanner.close();
26 }
27 // Función que verifica si un número es perfecto
28 public static boolean esPerfecto(int numero) {
29     int sumaDivisores = 0;
30     // Encontrar los divisores del número (excepto él mismo)
31     for (int i = 1; i < numero; i++) {
32         if (numero % i == 0) {
33             sumaDivisores += i; // Sumar los divisores
34         }
35     }
36     // Si la suma de los divisores es igual al número, es perfecto

```

```

37 }           return sumaDivisores == numero;
38 }
39 }
```

Captura de resultado:

```

run:
Ingrese un numero natural positivo: 16
El numero 16 no es perfecto.
BUILD SUCCESSFUL (total time: 3 seconds)
"run:
Ingrese un numero natural positivo: 28
El numero 28 es perfecto.
BUILD SUCCESSFUL (total time: 18 seconds)
```

17. Escribir un algoritmo que calcule y saque por pantalla los salarios de todos los empleados de una empresa. Cada empleado se identificará por un número desde el 1 hasta el total de empleados en la empresa. El salario es función de las horas trabajadas, que serán distintas para cada trabajador y se introducirán por teclado. El algoritmo aceptará también por teclado la cantidad que la empresa paga por hora ordinaria. Cuando un trabajador trabaje más de 40 horas, la empresa pagará las horas extras a 1.5 veces la hora ordinaria. Sugerencia: usar las variables ptas_hora, horas_trabajadas, salario, contador, y las constantes NumEmpleados=50 y FactorExtra=1.5

Algoritmo:

Inicio

1. Mostrar: "Ingrese el pago por hora ordinaria:"
2. Leer ptas_hora
3. Para contador \leftarrow 1 hasta NumEmpleados hacer:
 - Mostrar: "Ingrese las horas trabajadas por el empleado [contador]:"
 - Leer horas_trabajadas
 - Si horas_trabajadas $>$ 40 entonces:
 - horas_extra \leftarrow horas_trabajadas - 40
 - salario \leftarrow $(40 \times \text{ptas_hora}) + (\text{horas_extra} \times \text{ptas_hora} \times \text{FactorExtra})$
4. Sino:
 - salario \leftarrow horas_trabajadas \times ptas_hora
5. Mostrar: "El salario del empleado [contador] es: [salario]"

6. Mostrar: línea separadora

Fin

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class DecimoSeptimo {
6     /*
7         Escribir un algoritmo que calcule y saque por pantalla los salarios de todos los empleados de una
8         empresa. Cada empleado se identificará por un número desde el 1 hasta el total de empleados en la
9         empresa. El salario es función de las horas trabajadas, que serán distintas para cada trabajador y se
10        introducirán por teclado. El algoritmo aceptará también por teclado la cantidad que la empresa paga por
11        hora ordinaria. Cuando un trabajador trabaje más de 40 horas, la empresa pagará las horas extras a 1.5
12        veces la hora ordinaria. Sugerencia: usar las variables ptas_hora, horas_trabajadas, salario,
13        contador, y las constantes NumEmpleados=50 y FactorExtra=1'5
14     */
15
16     // Definimos constantes
17     public static final int NumEmpleados = 50;
18     public static final double FactorExtra = 1.5;
19     public static void main(String[] args) {
20         Scanner scanner = new Scanner(System.in);
21         double ptas_hora; // Valor por hora ordinaria
22
23         // Solicitar el valor por hora ordinaria
24         System.out.print("Ingrese el pago por hora ordinaria: ");
25         ptas_hora = scanner.nextDouble();
26
27         // Iterar por cada empleado
28         for (int contador = 1; contador <= NumEmpleados; contador++) {
29             System.out.print("Ingrese las horas trabajadas por el empleado " + contador + ": ");
30             double horas_trabajadas = scanner.nextDouble();
31             double salario;
32
33             // Si trabajó más de 40 horas, se pagan horas extras
34             if (horas_trabajadas > 40) {
35                 double horas_extra = horas_trabajadas - 40;
36                 salario = (40 * ptas_hora) + (horas_extra * ptas_hora * FactorExtra);
37             } else {
38                 salario = horas_trabajadas * ptas_hora;
39             }
40
41             // Mostrar salario
42             System.out.println("El salario del empleado " + contador + " es: " + salario);
43             System.out.println("-----");
44         }
45         scanner.close();
46     }
47 }
```

Captura de resultado:

```

run:
Ingrese el pago por hora ordinaria: 50
Ingrese las horas trabajadas por el empleado 1: 50
El salario del empleado 1 es: 2750.0
-----
Ingrese las horas trabajadas por el empleado 2: 30
El salario del empleado 2 es: 1500.0
-----
Ingrese las horas trabajadas por el empleado 3: 60
El salario del empleado 3 es: 3500.0
-----
Ingrese las horas trabajadas por el empleado 4: 42
El salario del empleado 4 es: 2150.0
-----
Ingrese las horas trabajadas por el empleado 5: 36
El salario del empleado 5 es: 1800.0
-----
Ingrese las horas trabajadas por el empleado 6: 29
El salario del empleado 6 es: 1450.0
-----
```

18. Escribir un algoritmo que saque en cada linea el cuadrado, el cubo y la raiz cuadrada de los P primeros numeros naturales. (Considerar predefinida una funcion RzCuadrada(num)).

Algoritmo:

Inicio

1. Definir P, i, cuadrado, cubo como Entero
2. Definir raiz como Real
3. Escribir "Ingrese la cantidad de numeros naturales (P):"

Leer P
4. Escribir "Numero | Cuadrado | Cubo | Raiz Cuadrada"
5. Escribir "-----"
6. Para i \leftarrow 1 Hasta P Con Paso 1 Hacer

cuadrado \leftarrow i \times i

cubo \leftarrow i \times i \times i

raiz \leftarrow RaizCuadrada(i)
7. Escribir i, cuadrado, cubo, raiz
8. FinPara

Fin

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class DecimoOctavo {
4     /*Escribir un algoritmo que saque en cada línea el cuadrado, el cubo y la raíz cuadrada de los P
5 primeros números naturales. (Considerar predefinida una función RzCuadrada(num)).*/
6     /*
7     // Función para calcular la raíz cuadrada
8     public static double RzCuadrada(int num) {
9         return Math.sqrt(num);
10    }
11    public static void main(String[] args) {
12        Scanner scanner = new Scanner(System.in);
13        // Leer la cantidad de números naturales
14        System.out.print("Ingrese la cantidad de numeros naturales (P): ");
15        int P = scanner.nextInt();
16        System.out.println("\nNúmero | Cuadrado | Cubo | Raiz Cuadrada");
17        System.out.println("-----");
18        // Iterar del 1 hasta P
19        for (int i = 1; i <= P; i++) {
20            int cuadrado = i * i;
21            int cubo = i * i * i;
22            double raiz = RzCuadrada(i);
23            // Mostrar los resultados
24            System.out.printf("%6d | %8d | %4d | %14.4f\n", i, cuadrado, cubo, raiz);
25        }
26    }
27 }

```

Captura de resultado:

```

run:
Ingrese la cantidad de numeros naturales (P): 10

Número | Cuadrado | Cubo | Raiz Cuadrada
-----
1 | 1 | 1 | 1,0000
2 | 4 | 8 | 1,4142
3 | 9 | 27 | 1,7321
4 | 16 | 64 | 2,0000
5 | 25 | 125 | 2,2361
6 | 36 | 216 | 2,4495
7 | 49 | 343 | 2,6458
8 | 64 | 512 | 2,8284
9 | 81 | 729 | 3,0000
10 | 100 | 1000 | 3,1623
BUILD SUCCESSFUL (total time: 3 seconds)

```

19. Escribir un algoritmo para un cajero automático que calcule el número óptimo de billetes de 20, 10, 5 y 1 para una cantidad entera de dinero en euros.

Algoritmo:

Inicio

1. Crear una variable cantidad para almacenar el monto total ingresado por el usuario.
2. Pedir al usuario que ingrese una cantidad entera en euros.
3. Leer y guardar ese valor en la variable cantidad.
4. Cálculo del desglose óptimo de billetes:

- Calcular cuántos billetes de 20€ caben en cantidad:
billetes20 = cantidad ÷ 20
- Actualizar la cantidad restante:
cantidad = cantidad módulo 20
- Calcular cuántos billetes de 10€ caben en la nueva cantidad:
billetes10 = cantidad ÷ 10
- Actualizar la cantidad restante:
cantidad = cantidad módulo 10
- Calcular cuántos billetes de 5€ caben en la nueva cantidad:
billetes5 = cantidad ÷ 5
- Actualizar la cantidad restante:
cantidad = cantidad módulo 5
- Los billetes de 1€ corresponden al resto que queda:
billetes1 = cantidad

5. Mostrar el número de billetes de cada denominación:

- Billetes de 20€, 10€, 5€ y 1€

Fin

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class DecimoNoveno {
4     /* Escribir un algoritmo para un cajero automático que calcule el número óptimo de billetes de 20, 10, 5
5 y 1 para una cantidad entera de dinero en euros.
6 */
7     public static void main(String[] args) {
8         Scanner scanner = new Scanner(System.in);
9         // Solicitar la cantidad de dinero al usuario
10        System.out.print("Ingrese la cantidad de dinero en euros: ");
11        int cantidad = scanner.nextInt();
12        // Calcular el número óptimo de billetes
13        int billetes20 = cantidad / 20;
14        cantidad %= 20;
15        int billetes10 = cantidad / 10;
16        cantidad %= 10;
17        int billetes5 = cantidad / 5;
18        cantidad %= 5;
19        int billetes1 = cantidad; // Lo que quede son billetes de 1€
20        // Mostrar los resultados
21        System.out.println("\nDesglose óptimo de billetes:");
22        System.out.println("Billetes de 20 euros: " + billetes20);
23        System.out.println("Billetes de 10 euros: " + billetes10);
24        System.out.println("Billetes de 5 euros : " + billetes5);
25        System.out.println("Billetes de 1 euros : " + billetes1);
26        scanner.close();
27    }
28 }
```

Captura de resultado:

```

run:
Ingrese la cantidad de dinero en euros: 856

Desglose optimo de billetes:
Billetes de 20 euros: 42
Billetes de 10 euros: 1
Billetes de 5 euros : 1
Billetes de 1 euros : 1
BUILD SUCCESSFUL (total time: 3 seconds)

```

20. Se definen los números triangulares como los obtenidos por la suma de los números naturales sucesivos 1, 2, 3, Es decir, los primeros números triangulares son 1, 3, 6, 10, etc. Elabore un programa para imprimir el n-ésimo número triangular.

Algoritmo:

Inicio

1. Crear una variable llamada n que representará la posición del número triangular a calcular.
2. Solicitar al usuario que ingrese un valor entero para n.
3. Leer y almacenar ese valor en la variable n.
4. Calcular el n-ésimo número triangular usando la fórmula:

$$\frac{n(n + 1)}{2}$$

5. Guardar ese valor en la variable numeroTriangular.
6. Mostrar el valor de numeroTriangular junto con un mensaje indicando que es el n-ésimo número triangular.

Fin

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class Vigesimo {
6
7     /* Se definen los números triangulares como los obtenidos por la suma de los números naturales
8     sucesivos 1, 2, 3, ... Es decir, los primeros números triangulares son 1, 3, 6, 10, etc. Elabore un programa
9     para imprimir el n-ésimo número triangular.
10    */
11
12     public static void main(String[] args) {
13
14         Scanner scanner = new Scanner(System.in);
15
16         // Pedir el valor de n
17         System.out.print("Ingrese el valor de n: ");
18         int n = scanner.nextInt();
19
20         // Calcular el n-ésimo número triangular
21         int numeroTriangular = n * (n + 1) / 2;
22
23         // Mostrar el resultado
24         System.out.println("El " + n + "-ésimo numero triangular es: " + numeroTriangular);
25
26         scanner.close();
27     }
28 }
```

Captura de resultado:

```

Ingrese el valor de n: 5
El 5-ésimo numero triangular es: 15

```

21. Elabore un programa que imprima el triángulo de Floyd hasta un valor dado. El triángulo contiene los números naturales correlativos, uno en la primera línea, dos en la segunda, etc.; es decir, en la fila enesima aparecen n valores.

Algoritmo:

Inicio

1. Crear una variable llamada filas para almacenar la cantidad de filas que tendrá el triángulo.
2. Solicitar al usuario que ingrese el número de filas del triángulo.
3. Leer y almacenar ese valor en la variable filas.
4. Inicializar un ciclo externo que irá desde $i = 1$ hasta $i = \text{filas}$, y que controlará las filas.
5. Dentro del ciclo externo, crear un ciclo interno que irá desde $j = 1$ hasta $j = i$, para imprimir i números por cada fila.
6. En cada iteración del ciclo interno, imprimir el valor de j seguido de un espacio.
7. Al finalizar cada fila, imprimir un salto de línea para pasar a la siguiente fila.

8. Repetir el proceso hasta completar todas las filas.

Fin

Código fuente:

```

1 package Grupo1Problemas;
2
3 import java.util.Scanner;
4
5 public class VigesimoPrimer {
6
7     /* Elabore un programa que imprima el triángulo de Floyd hasta un valor dado. El triángulo contiene
8     los números naturales correlativos, uno en la primera línea, dos en la segunda, etc.; es decir,
9     en la fila nesima aparecen n valores.
10 */
11
12     public static void main(String[] args) {
13
14         Scanner scanner = new Scanner(System.in);
15
16         // Pedir al usuario el número de filas del triángulo
17         System.out.print("Ingrese el numero de filas del triangulo: ");
18         int filas = scanner.nextInt();
19
20         // Imprimir el triángulo
21         for (int i = 1; i <= filas; i++) {
22             for (int j = 1; j <= i; j++) {
23                 System.out.print(j + " ");
24             }
25             System.out.println();
26         }
27
28         scanner.close();
29     }
30 }
```

Captura de resultado:

```

Ingrese el numero de filas del triangulo: 10
1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
BUILD SUCCESSFUL (total time: 3 seconds)
```

22. Generar la tabla para "N" varía entre 1.0 y 1.1 con incremento de 0.001. la tabla

N N² N^{0,5}

99 9999 999

Algoritmo:

Inicio del programa
1. Comenzar el programa.

2. Mostrar la cabecera de la tabla con tres columnas:

- "N" (valor original).
- "N²" (cuadrado de N).
- " \sqrt{N} " (raíz cuadrada de N).

3. Inicializar n = 1.0.

4. Mientras n ≤ 1.1, hacer lo siguiente:

- Calcular cuadrado = n × n.
- Calcular raíz = \sqrt{n} .
- Mostrar n, cuadrado y raíz con formato decimal:
 - n con 3 decimales.
 - cuadrado y raíz con 6 decimales.
- Incrementar n en 0.001.

Fin

Código fuente:

```

1 package Grupo1Problemas;
2
3
4
5 public class VigesimoSegundo {
6
7     /*Generar la tabla para "N" varía entre 1.0 y 1.1 con incremento de 0.001. la tabla
8         N          N^2          N^0,5
9         99         9999        999
10    */
11    public static void main(String[] args) {
12
13        System.out.printf("%-10s%-10s%-10s\n", "N", "N^2", " $\sqrt{N}$ ");
14
15        for (double n = 1.0; n <= 1.1; n += 0.001) {
16            double cuadrado = n * n;
17            double raiz = Math.sqrt(n);
18            System.out.printf("%.3f    %.6f    %.6f\n", n, cuadrado, raiz);
19        }
20    }
21 }
```

Captura de resultado:

--

run:	N	N^2	?N
	1,000	1,000000	1,000000
	1,001	1,002001	1,000500
	1,002	1,004004	1,001000
	1,003	1,006009	1,001499
	1,004	1,008016	1,001998
	1,005	1,010025	1,002497
	1,006	1,012036	1,002996
	1,007	1,014049	1,003494
	1,008	1,016064	1,003992
	1,009	1,018081	1,004490
	1,010	1,020100	1,004988
	1,011	1,022121	1,005485
	1,012	1,024144	1,005982
	1,013	1,026169	1,006479
	1,014	1,028196	1,006976
	1,015	1,030225	1,007472
	1,016	1,032256	1,007968
	1,017	1,034289	1,008464
	1,018	1,036324	1,008960
	1,019	1,038361	1,009455
	1,020	1,040400	1,009950
	1,021	1,042441	1,010445
	1,022	1,044484	1,010940
	1,023	1,046529	1,011435
	1,024	1,048576	1,011929
	1,025	1,050625	1,012423
	1,026	1,052676	1,012917
	1,027	1,054729	1,013410

23. Un matemático descubre un método para multiplicar cualquier número entero por 11 mediante la suma de sus dígitos. Sea un número entero N que se expresa con M dígitos, el producto de dicho número por 11 esta dado por la siguiente regla: “el dígito de las unidades es igual al dígito de las unidades de N, el dígito de las decenas es igual a la suma de los dígitos de unidades y decenas de N, el dígito de las centenas es igual a la suma de los dígitos de decenas y centenas de N, y así sucesivamente. Ejemplo: el producto del número 3247 por 11 es 35717. Escribir un algoritmo para que pueda multiplicar cualquier número entero por 11

Algoritmo:

1. Entrada:
 - o Leer el número entero NNN, representado como una cadena de caracteres.
2. Inicialización:
 - o Convertir la cadena de caracteres NNN en un arreglo de dígitos.
 - o Crear un arreglo llamado resultado[] para almacenar el resultado de la multiplicación.
3. Proceso (aplicar la regla de multiplicación por 11):
 - o Recorrer el número NNN de derecha a izquierda:

- Primer dígito (más a la derecha):
El primer dígito de NNN se coloca tal cual en el resultado.
 - Dígitos intermedios (desde el penúltimo hasta el primero):
Para cada dígito i_{ii} (excepto el primero y el último):
 - La suma de los dígitos i_{ii} y $i_{i-1}i_{i-1}i_{i-1}$ de NNN se coloca en la posición i_{ii} de resultado[].
 - Si la suma es mayor o igual a 10, se maneja el acarreo:
 - Restar 10 al valor en la posición i_{ii} .
 - Sumar 1 al dígito anterior $i_{i-1}i_{i-1}i_{i-1}$ para reflejar el acarreo.
 - Último dígito (más a la izquierda):
Se agrega directamente al arreglo resultado[].
4. Acarreo:
- Si en cualquier posición el valor de resultado[i] es mayor o igual a 10, se maneja el acarreo, sumando 1 a la posición anterior y ajustando el valor actual para que quede entre 0 y 9.
5. Salida:
- Imprimir el arreglo resultado[] sin los ceros iniciales, es decir, omitir cualquier valor de 0 al principio.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Scanner;
3 public class VigesimoTercero {
4     public static void main(String[] args) {
5         Scanner sc = new Scanner(System.in);
6         System.out.print("Ingresa un numero entero para multiplicarlo por 11: ");
7         String numero = sc.nextLine();
8         // Convertir el número en un arreglo de caracteres
9         int[] digitos = new int[numero.length() + 1]; // +1 para el resultado
10        for (int i = 0; i < numero.length(); i++) {
11            digitos[i + 1] = numero.charAt(i) - '0'; // Desfase por el inicio en 1
12        }
13        // Arreglo para el resultado
14        int[] resultado = new int[digitos.length + 1]; // +1 por posible acarreo al inicio
15        // Aplicar la regla: sumar digitos adyacentes
16        for (int i = digitos.length - 1; i >= 1; i--) {
17            resultado[i] += digitos[i] + digitos[i - 1];
18            // Manejo del acarreo
19            if (resultado[i] >= 10) {
20                resultado[i] -= 10;
21                resultado[i - 1] += 1;
22            }
23        }
24        // Agregar el último dígito (unidades)
25        resultado[resultado.length - 1] = digitos[digitos.length - 1];
26
27        // Imprimir el resultado (saltando ceros al inicio)
28        System.out.print("Resultado de multiplicar por 11: ");
29        boolean inicio = false;
30        for (int i = 0; i < resultado.length; i++) {
31            if (!inicio && resultado[i] == 0) continue;
32            inicio = true;
33            System.out.print(resultado[i]);
34        }
35    }
36 }
37

```

Captura de resultado:

```

Ingresa un numero entero para multiplicarlo por 11: 60
Resultado de multiplicar por 11: 660

```

24. La escuela Académica Profesional de Ingeniería de Sistemas quiere realizar los algoritmos necesarios para realizar la evaluación y selección de un conjunto de mil postulantes para el ingreso de admisión para el Programa de Graduación de Ingeniería de Sistemas, considerando los siguientes parámetros:

- ◆ Número de vacantes: 150.
- ◆ Número de preguntas es 100.
- ◆ La nota mínima de ingreso es 120 puntos.
- ◆ El examen consta de tres partes: Las 40 primeras preguntas de Razonamiento matemático y verbal (2 puntos correcto, -1 si es incorrecto), 30 preguntas de conocimiento en ciencias (3 puntos correcto, -1.5 si es incorrecto), y 30 preguntas de conocimiento en letras (1 punto correcto, -0.5 si es incorrecto).
 - Calcule la nota final del postulante y determine si ingreso o no ingreso el postulante.
 - Calcule la media aritmética de las notas de todos los postulantes.

c) Calcule la varianza de las notas de los postulantes.

d) Calcule la nota mínima y máxima de los ingresantes.

Algoritmo:

1. Entrada:

- Leer el número entero NNN, representado como una cadena de caracteres.

2. Inicialización:

- Convertir la cadena de caracteres NNN en un arreglo de dígitos.
- Crear un arreglo llamado resultado[] para almacenar el resultado de la multiplicación.

3. Proceso (aplicar la regla de multiplicación por 11):

- Recorrer el número NNN de derecha a izquierda:
 - Primer dígito (más a la derecha):
 - El primer dígito de NNN se coloca tal cual en el resultado.
 - Dígitos intermedios (desde el penúltimo hasta el primero):
 - Para cada dígito i (excepto el primero y el último):
 - La suma de los dígitos i y $i-1$ de NNN se coloca en la posición i de resultado[].
 - Si la suma es mayor o igual a 10, se maneja el acarreo:
 - Restar 10 al valor en la posición i .
 - Sumar 1 al dígito anterior $i-1$ para reflejar el acarreo.
 - Último dígito (más a la izquierda):
 - Se agrega directamente al arreglo resultado[].

4. Acarreo:

- Si en cualquier posición el valor de resultado[i] es mayor o igual a 10, se maneja el acarreo, sumando 1 a la posición anterior y ajustando el valor actual para que quede entre 0 y 9.

5. Salida:

- Imprimir el arreglo resultado[] sin los ceros iniciales, es decir, omitir cualquier valor de 0 al principio.

Código fuente:

```

1 package Grupo1Problemas;
2 import java.util.Random;
3 import java.util.Arrays;
4 public class VigesimoCuarto {
5     public static void main(String[] args) {
6         final int NUM_POSTULANTES = 1000;
7         final int VACANTES = 150;
8         final double NOTA_MIN_INGRESO = 120;
9         double[] notas = new double[NUM_POSTULANTES];
10        int ingresantes = 0;
11        Random rand = new Random();
12        for (int i = 0; i < NUM_POSTULANTES; i++) {
13            int correctasRMV = rand.nextInt(41);
14            int incorrectasRMV = 40 - correctasRMV;
15            int correctasCiencias = rand.nextInt(31);
16            int incorrectasCiencias = 30 - correctasCiencias;
17            int correctasLetras = rand.nextInt(31);
18            int incorrectasLetras = 30 - correctasLetras;
19            double nota = (2 * correctasRMV - 1 * incorrectasRMV) +
20                (3 * correctasCiencias - 1.5 * incorrectasCiencias) +
21                (1 * correctasLetras - 0.5 * incorrectasLetras);
22            notas[i] = nota;
23            if (nota >= NOTA_MIN_INGRESO) {
24                ingresantes++;
25            }
26        }
27        Arrays.sort(notas);
28        double[] notasIngresantes = Arrays.copyOfRange(notas, NUM_POSTULANTES - VACANTES, NUM_POSTULANTES);
29        double sumaNotas = 0;
30        for (double nota : notas) {
31            sumaNotas += nota;
32        }
33        double media = sumaNotas / NUM_POSTULANTES;
34        double sumaVarianza = 0;
35        for (double nota : notas) {
36            sumaVarianza += Math.pow(nota - media, 2);
37        }
38        double varianza = sumaVarianza / NUM_POSTULANTES;
39        double notaMin = notasIngresantes[0];
40        double notaMax = notasIngresantes[notasIngresantes.length - 1];
41        System.out.println("\nResultados simulados:");
42        System.out.println("Cantidad total de ingresantes (nota >= 120): " + ingresantes);
43        System.out.println("Media de notas: " + media);
44        System.out.println("Varianza de notas: " + varianza);
45        System.out.println("Nota minima de los 150 mejores: " + notaMin);
46        System.out.println("Nota maxima de los 150 mejores: " + notaMax);
47    }
48 }

```

Captura de resultado:

```

Resultados simulados:
Cantidad total de ingresantes (nota >= 120): 134
Media de notas: 51.4175
Varianza de notas: 3134.6729437500007
Nota minima de los 150 mejores: 116.0
Nota maxima de los 150 mejores: 189.5

```


EJERCICIOS PROPUESTOS (Python)

Elaborado por: Avalos Palomino Ismael Jesús

- Ejercicio 1

```

py ejercicio_14.py ejercicio_15.py ejercicio_1.py ejercicio_16.py ejercicio_17.py ejercicio_18.py ejercicio_19.py ejercicio_20.py D ...
TAREA 1 > ejercicio_1.py > ...
1 print("Tabla del 3")
2 for i in range(1, 11):
3     print(f"3 x {i} = {3 * i}")
4
5 print("\nTabla del 9")
6 for i in range(1, 11):
7     print(f"9 x {i} = {9 * i}")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS + ... x
+ Python ▲
+ Python ▲
+ Python ▲
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 = 30
Tabla del 9
9 x 1 = 9
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 = 90
PS C:\Users\Usuario\Desktop\Python> []

```

- Ejercicio 2

```

ejercicio_2.py X ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py D ...
TAREA 1 > ejercicio_2.py > ...
1 numero = int(input("Ingresa un número entero: "))
2 suma_pares = 0
3 suma_impar = 0
4 suma_multiplos_5 = 0
5
6 for i in range(1, numero + 1):
7     if i % 2 == 0:
8         suma_pares += i
9     else:
10        suma_impar += i
11    if i % 5 == 0:
12        suma_multiplos_5 += i
13
14 print(f"Suma de números pares desde 1 hasta {numero}: {suma_pares}")
15 print(f"Suma de números impares desde 1 hasta {numero}: {suma_impar}")
16 print(f"Suma de múltiplos de 5 desde 1 hasta {numero}: {suma_multiplos_5}")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS + ... x
+ Python + ... x
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_2.py"
Ingresa un número entero: 25
Suma de números pares desde 1 hasta 25: 156
Suma de números impares desde 1 hasta 25: 169
Suma de múltiplos de 5 desde 1 hasta 25: 75
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 3

The screenshot shows a code editor interface with a dark theme. At the top, there's a navigation bar with tabs like PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL, and PUERTOS. Below the tabs, the terminal window displays the following text:

```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 1/ejercicio_3.py"
La suma de los 20 primeros multiplos de 3 es 60
PS C:\Users\Usuario\Desktop\Phyton>
```

The code itself is a simple script named `ejercicio_3.py` that calculates the sum of the first 20 multiples of 3:

```
TAREA 1 > ejercicio_3.py > ...
1 sum=0
2 for i in range(1,21):
3     multiplo=i*3
4     suma = multiplo
5 print("La suma de los 20 primeros multiplos de 3",suma)
6
7
```

- Ejercicio 4

The screenshot shows a code editor interface with a dark theme. At the top, there's a navigation bar with tabs like PROBLEMAS, SALIDA, CONSOLA DE DEPURACIÓN, TERMINAL, and PUERTOS. Below the tabs, the terminal window displays the following text:

```
PS C:\Users\Usuario\Desktop\Phyton> & c:/Users/Usuario/Desktop/Phyton/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Phyton/TAREA 1/ejercicio_4.py"
resultado: 45
PS C:\Users\Usuario\Desktop\Phyton>
```

The code itself is a script named `ejercicio_4.py` that performs a series of operations on a number, including rotation and averaging:

```
TAREA 1 > ejercicio_4.py > ...
1 def rotar_izquierda(num_str):
2     """Un numero que rota a la izquierda, representa como cadena"""
3     return num_str[1:] + num_str[0]
4
5 def promedio_digital(numero):
6     """Calculamos los dígitos del número"""
7     digitos=[int(d) for d in str(numero)]
8     promedio=sum(digitos)/len(digitos)
9     return round(promedio)
10
11 def rotar_por_promedio(numero):
12     num_str=str(numero)
13     rotaciones=promedio_digital(numero)
14
15     for _ in range(rotaciones):
16         num_str=rotar_izquierda(num_str)
17
18     return int(num_str)
19 #ejemplo de el uso
20 numero=345
21 resultado=rotar_por_promedio(numero)
22 print("resultado:", resultado)
23
```

- Ejercicio 5

```

ejercicio_2.py ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py
TAREA 1 > ejercicio_5.py ...
1 contador_total = 0
2 contador_bisiestos = 0
3 print("Ingresá hasta 10 años. El programa terminará si se ingresan 3 años bisiestos.")
4 while contador_total < 10 and contador_bisiestos < 3:
5     try:
6         año = int(input("Ingresá el año #[contador_total + 1]: "))
7
8         if (año % 4 == 0 and año % 100 != 0) or (año % 400 == 0):
9
10             print(f"{año} es un año bisiesto.")
11             contador_bisiestos += 1
12         else:
13             print(f"{año} NO es un año bisiesto.")
14
15     contador_total += 1
16
17 except ValueError:
18     print("Por favor, ingresa un número entero válido.")
19
20 print("\nPrograma finalizado.")
21 print(f"Años ingresados: {contador_total}")
22 print(f"Años bisiestos encontrados: {contador_bisiestos}")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
Ingrese el año #6: 220
220 es un año bisiesto.
Ingrese el año #7: 2024
214 NO es un año bisiesto.
Ingrese el año #6: 220
220 es un año bisiesto.
Ingrese el año #7: 2024
Ingrese el año #7: 2024
2024 es un año bisiesto.

Programa finalizado.
Años ingresados: 7
2024 es un año bisiesto.

Programa finalizado.
Años ingresados: 7
Años bisiestos encontrados: 3
PS C:\Users\Usuario\Desktop\Python> []

```

- Ejercicio 6

```

ejercicio_2.py ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py
TAREA 1 > ejercicio_6.py ...
1 import math
2 def aproximar_seno(x_rad, n=10):
3     seno = 0
4     for i in range(n):
5         signo = (-1)**i
6         termino = signo * (x_rad ** (2 * i + 1)) / math.factorial(2 * i + 1)
7         seno += termino
8     return seno
9
10 try:
11     angulo = float(input("Ingresá un ángulo en grados (entre 0 y 360): "))
12
13     if 0 <= angulo <= 360:
14         radianes = math.radians(angulo)
15
16         seno_aprox = aproximar_seno(radianes)
17
18         print(f"Seno aproximado de {angulo}: {seno_aprox:.6f}")
19
20         print(f"Seno real (math.sin): {math.sin(radianes):.6f}")
21     else:
22         print("Error: el ángulo debe estar entre 0 y 360 grados.")
23
24 except ValueError:
25     print("Error: debes ingresar un número válido.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_6.py"
Ingresá un ángulo en grados (entre 0 y 360): 270
Seno aproximado de 270.0: -1.000000
Seno real (math.sin): -1.000000
PS C:\Users\Usuario\Desktop\Python> []

```

- Ejercicio 7

```

TAREA 1 > ejercicio_7.py > ...
1 import random
2 maximo=100 #constante
3 #Generar un numero aleatorio del 1 y maximo
4 numero=random.randint(1,maximo)
5 acierto=False
6 intentos=0
7 while not acierto:
8     n=int(input("Introduzca un número:"))
9     intentos+=1
10    if n==numero:
11        print("Acertaste!!")
12        acierto=True
13    elif n<numero:
14        print("El numero es mayor")
15    else:
16        print("El numero es menor")
17
18 print(f"Adivinaste el numero en {intentos} intento(s).")
20

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_7.py"

Introduzca un número:5
El numero es menor
Introduzca un número:4
El numero es menor
Introduzca un número:7
El numero es menor
Introduzca un número:1
El numero es mayor
Introduzca un número:5
El numero es menor
Introduzca un número:[]

- Ejercicio 8

```

TAREA 1 > ejercicio_8.py > ...
1 numero=int(input("Insertar numero:"))#leer numero desde la entrada estandar
2
3 #imprimir la tabla de multiplicar
4 print(f"Tabla de multiplicar del{numero}:")
5 for i in range(1,11):
6     resultado = numero*i
7     print(f"{numero}x{i}={resultado}")
8

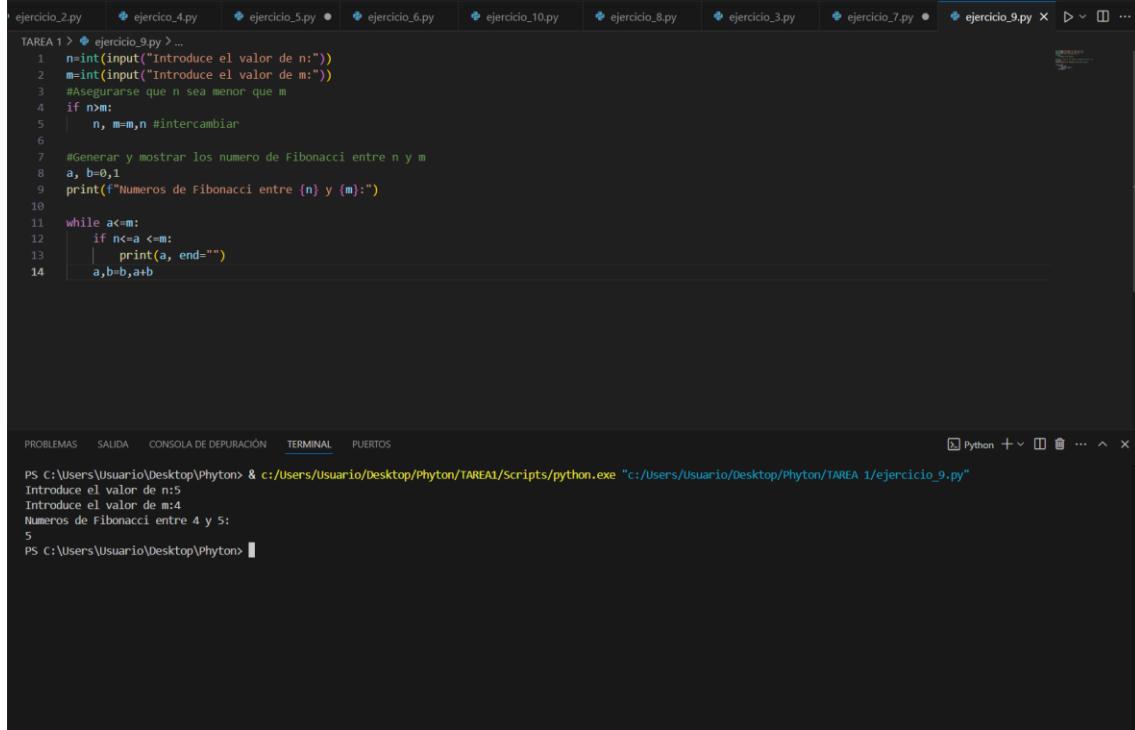
```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_8.py"

Insertar numero:5
Tabla de multiplicar del5:
5x1=5
5x2=5
5x3=5
5x4=5
5x5=5
5x6=5
5x7=5
5x8=5
5x9=5
5x10=5
PS C:\Users\Usuario\Desktop\Python>

- Ejercicio 9



```

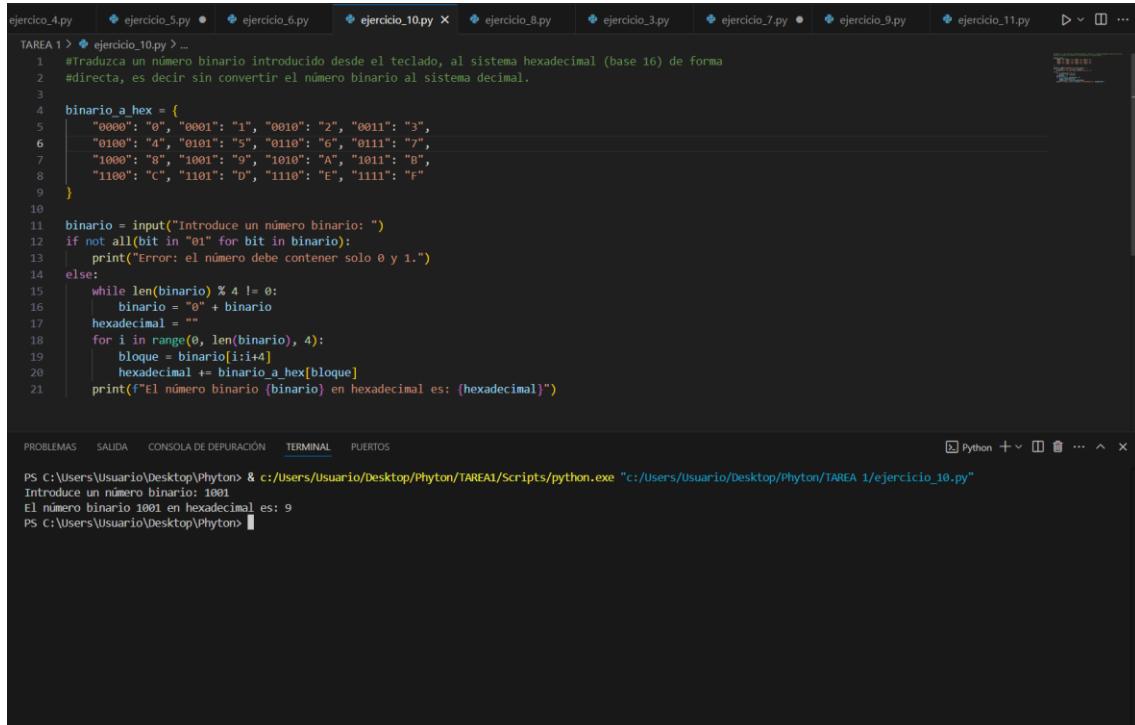
ejercicio_2.py ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py
TAREA1 > ejercicio_9.py > ...
1 n=int(input("Introduce el valor de n:"))
2 m=int(input("Introduce el valor de m:"))
3 #Asegurarse que n sea menor que m
4 if n>m:
5     n, m=m,n #intercambiar
6
7 #Generar y mostrar los numero de Fibonacci entre n y m
8 a, b=0,1
9 print(f"Numeros de Fibonacci entre {n} y {m}:")
10
11 while a<=m:
12     if n<=a <=m:
13         print(a, end="")
14     a,b=b,a+b

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_9.py"
Introduce el valor de n:5
Introduce el valor de m:4
Numeros de Fibonacci entre 4 y 5:
5

- Ejercicio 10



```

ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py ejercicio_11.py
TAREA1 > ejercicio_10.py > ...
1 #Traduzca un número binario introducido desde el teclado, al sistema hexadecimal (base 16) de forma
2 #directa, es decir sin convertir el número binario al sistema decimal.
3
4 binario_a_hex = {
5     "0000": "0", "0001": "1", "0010": "2", "0011": "3",
6     "0100": "4", "0101": "5", "0110": "6", "0111": "7",
7     "1000": "8", "1001": "9", "1010": "A", "1011": "B",
8     "1100": "C", "1101": "D", "1110": "E", "1111": "F"
9 }
10
11 binario = input("Introduce un número binario: ")
12 if not all(bit in "01" for bit in binario):
13     print("Error: el número debe contener solo 0 y 1.")
14 else:
15     while len(binario) % 4 != 0:
16         binario = "0" + binario
17     hexadecimal = ""
18     for i in range(0, len(binario), 4):
19         bloque = binario[i:i+4]
20         hexadecimal += binario_a_hex[bloque]
21     print(f"El número binario {binario} en hexadecimal es: {hexadecimal}")

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_10.py"
Introduce un número binario: 1001
El número binario 1001 en hexadecimal es: 9

- Ejercicio 11

```

ejercicio_4.py ejercicio_5.py ejercicio_6.py ejercicio_10.py ejercicio_8.py ejercicio_3.py ejercicio_7.py ejercicio_9.py ejercicio_11.py ... ...
TAREA 1 > ejercicio_11.py > ...
1 total=0
2 acepta_a=0
3 acepta_b=0
4 ambos=0
5 solo_a=0
6 solo_b=0
7 ninguno=0
8 print("Encuenta de aceptacion de productos a y b")
9 print("Ingresa los ddatos como pares(a,b) usando 1 o 0.Escribe'fin'para terminar.\n")
10 while True:
11     entrada=input("Par(a,b):")
12     if entrada.lower() == "fin":
13         break
14     try:
15         a,b=map(int,entrada.strip().split(","))
16         if a not in (0,1) or b not in (0,1):
17             print("Valores invalidos. Usa solo 0 o 1.")
18             continue
19         total+=1
20         if a==1:
21             acepta_a+=1
22         if b==1:
23             acepta_b+=1
24         if a==1 and b==1:
25             ambos+=1
26         elif a==1 and b==0:
27             solo_a+=1
28         elif a==0 and b==1:
29             solo_b+=1
30         elif a==0 and b==0:
31             ninguno+=1
32     except:
33         print("Formato invalido. Usa el formato(1,0), (0,1), etc.")
34     if total > 0:
35         print(f"\nTotal de encuestados: {total}")
36         print(f"Porcentaje que aceptan el producto A: {acepta_a / total * 100:.2f}%")
37         print(f"Porcentaje que aceptan el producto B: {acepta_b / total * 100:.2f}%")
38         print(f"Porcentaje que aceptan ambos productos: {ambos / total * 100:.2f}%")
39         print(f"Porcentaje que aceptan solo el producto A: {solo_a / total * 100:.2f}%")
40         print(f"Porcentaje que aceptan solo el producto B: {solo_b / total * 100:.2f}%")
41         print(f"Porcentaje que no aceptan ninguno: {ninguno / total * 100:.2f}%")
42     else:
43

```

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/users/Usuario/Desktop/Python/TAREA 1/ejercicio_11.py"
Encuenta de aceptacion de productos a y b
Ingresa los ddatos como pares(a,b) usando 1 o 0.Escribe'fin'para terminar.

Par(a,b):1,0
Par(a,b):0,1
Par(a,b):1,1
Par(a,b):0,0
Par(a,b):fin

Total de encuestados: 1
Porcentaje que aceptan el producto A: 100.00%
Porcentaje que aceptan el producto B: 100.00%
Porcentaje que aceptan ambos productos: 100.00%
Porcentaje que aceptan solo el producto A: 100.00%
Porcentaje que aceptan solo el producto B: 100.00%
Porcentaje que no aceptan ninguno: 100.00%
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 12

```
ejercicio_3.py ejercicio_7.py ejercicio_9.py ejercicio_11.py ejercicio_22.py ejercicio_23.py ejercicio_24.py ejercicio_25.py ejercicio_12.py x D v x

TAREA 1 > ejercicio_12.py > arabigo_a_romano
1 def arabigo_a_romano(numero):
2     romanos = {
3         1: "I", 2: "II", 3: "III", 4: "IV", 5: "V",
4         6: "VI", 7: "VII", 8: "VIII", 9: "IX", 10: "X",
5         11: "XI", 12: "XII", 13: "XIII", 14: "XIV", 15: "XV",
6         16: "XVI", 17: "XVII", 18: "XVIII", 19: "XIX", 20: "XX"
7     }
8     if 1 <= numero <= 20:
9         return romanos[numero]
10    else:
11        return "Número fuera de rango (1-20)"
12 try:
13     entrada = int(input("Ingrese un número del 1 al 20: "))
14     resultado = arabigo_a_romano(entrada)
15     print(f"Número romano: {resultado}")
16 except ValueError:
17     print("Entrada inválida. Por favor, ingrese un número entero.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS Python + v x

PS C:\Users\Usuario\Desktop\Python & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_12.py"
Ingrese un número del 1 al 20: 11
Número romano: XI
PS C:\Users\Usuario\Desktop\Python>
```

- Ejercicio 13

- Ejercicio 14

```

TAREA 1 > ⚡ ejercicio_14.py > ...
1 numeros = []
2 print("Ingrese números positivos uno por uno. Ingrese 0 para terminar.")
3 while True:
4     try:
5         n = int(input("Número: "))
6         if n < 0:
7             print("Solo se permiten números positivos.")
8         elif n == 0:
9             break
10        else:
11            numeros.append(n)
12    except ValueError:
13        print("Entrada inválida. Por favor ingrese un número entero.")
14 if len(numeros) == 0:
15     print("No se ingresaron números válidos.")
16 else:
17     menor = min(numeros)
18     mayor = max(numeros)
19     promedio = sum(numeros) / len(numeros)
20     print(f"Menor: {menor}")
21     print(f"Mayor: {mayor}")
22     print(f"Promedio: {promedio:.2f}")

PROBLEMAS SALIDA CONSOLA DE DÉPURAÇÃO TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_14.py"
Ingrese números positivos uno por uno. Ingrese 0 para terminar.
Número: 15
Número: 12
Número: 14
Número: 0
Menor: 12
Mayor: 15
Promedio: 13.67
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 15

```

11.py ⚡ ejercicio_22.py • ⚡ ejercicio_23.py ⚡ ejercicio_24.py ⚡ ejercicio_25.py ⚡ ejercicio_12.py ⚡ ejercicio_13.py ⚡ ejercicio_14.py ⚡ ejercicio_15.py X ▶ ▾ □ ...
TAREA 1 > ⚡ ejercicio_15.py > ⚡ fibonacci
1 def fibonacci(n):
2     if n <= 0:
3         print("El número debe ser natural y mayor que 0.")
4         return
5
6     f1, f2 = 1, 1
7     print("Sucesión de Fibonacci:")
8     for i in range(n):
9         print(f1, end=' ')
10        f1, f2 = f2, f1 + f2
11
12 # Programa principal
13 try:
14     n = int(input("Ingrese el número de términos N: "))
15     fibonacci(n)
16 except ValueError:
17     print("Entrada inválida. Ingrese un número entero natural.")

PROBLEMAS SALIDA CONSOLA DE DÉPURAÇÃO TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_15.py"
Ingrese el número de términos N:
Sucesión de Fibonacci:
1 1 2 3 5
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 16

```

ejercicio_24.py ejercicio_25.py ejercicio_12.py ejercicio_13.py ejercicio_14.py ejercicio_15.py ejercicio_1.py ejercicio_16.py
TAREA 1 > ejercicio_16.py > es_perfecto(n):
1     def es_perfecto(n):
2         if n <= 0:
3             return False
4         suma_divisores = 0
5         for i in range(1, n):
6             if n % i == 0:
7                 suma_divisores += i
8         return suma_divisores == n
9     try:# Programa principal
10        numero = int(input("Ingrese un número natural positivo: "))
11        if numero <= 0:
12            print("Debe ser un número natural positivo.")
13        else:
14            if es_perfecto(numero):
15                print(f"{numero} es un número perfecto.")
16            else:
17                print(f"{numero} no es un número perfecto.")
18    except ValueError:
19        print("Entrada inválida. Ingrese un número entero.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_16.py"
Ingrese un número natural positivo: 5
5 no es un número perfecto.
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 17

```

ejercicio_24.py ejercicio_25.py ejercicio_12.py ejercicio_13.py ejercicio_14.py ejercicio_15.py ejercicio_1.py ejercicio_16.py ejercicio_17.py
TAREA 1 > ejercicio_17.py > ...
1 NumEmpleados = 50
2 FactorExtra = 1.5
3 try:# solicitar tarifa por hora
4     ptas_hora = float(input("Ingrese la tarifa por hora ordinaria: "))
5
6 # Bucle para cada empleado
7 for contador in range(1, NumEmpleados + 1):
8     horas_trabajadas = float(input("Ingrese las horas trabajadas el empleado [contador]: "))
9
10    if horas_trabajadas <= 40:
11        salario = horas_trabajadas * ptas_hora
12    else:
13        horas_extra = horas_trabajadas - 40
14        salario = (40 * ptas_hora) + (horas_extra * ptas_hora * FactorExtra)
15
16    print(f"Empleado {contador} → Salario: {salario:.2f}")
17 except ValueError:
18     print("Entrada inválida. Asegúrese de ingresar valores numéricos.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_17.py"
Ingrese la tarifa por hora ordinaria: 50
Ingrese las horas trabajadas el empleado 1: 5
Empleado 1 → Salario: 250.00
Ingrese las horas trabajadas el empleado 2: 4
Empleado 2 → Salario: 200.00
Ingrese las horas trabajadas el empleado 3: 5
Empleado 3 → Salario: 250.00
Ingrese las horas trabajadas el empleado 4: 8
Empleado 4 → Salario: 400.00
Ingrese las horas trabajadas el empleado 5: 7
Empleado 5 → Salario: 350.00
Ingrese las horas trabajadas el empleado 6: 5
Empleado 6 → Salario: 250.00
Ingrese las horas trabajadas el empleado 7: 4
Empleado 7 → Salario: 200.00
Ingrese las horas trabajadas el empleado 8: 50
Empleado 8 → Salario: 2750.00
Ingrese las horas trabajadas el empleado 9: 1

```

- Ejercicio 18

```

ejercicio_18.py
1 import math
2 def RzCuadrada(num):
3     return math.sqrt(num)
4 try:
5     P = int(input("Ingrese el número P de primeros números naturales: "))
6     print(f"\n{P} {RzCuadrada(P)}")
7     print(f"\n{P} {P**2} {P**3} {RzCuadrada(P)}")
8     for i in range(1, P + 1):
9         cuadrado = i ** 2
10        cubo = i ** 3
11        raiz = RzCuadrada(i)
12        print(f"\n{i} {cuadrado} {cubo} {raiz}")
13    except ValueError:
14        print("Por favor, ingrese un número natural válido.")

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_18.py"

Ingrese el número P de primeros números naturales: 5

Nº	Cuadrado	Cubo	Raíz Cuadrada
1	1	1	1.0000
2	4	8	1.4142
3	9	27	1.7321
4	16	64	2.0000
5	25	125	2.2361

PS C:\Users\Usuario\Desktop\Python>

- Ejercicio 19

```

ejercicio_19.py
1 def calcular_billetes(monto):
2     billetes_20 = monto // 20
3     monto %= 20
4     billetes_10 = monto // 10
5     monto %= 10
6     billetes_5 = monto // 5
7     monto %= 5
8     billetes_1 = monto # Lo que quede son monedas de 1 €
9     return billetes_20, billetes_10, billetes_5, billetes_1
10
11 try:
12     cantidad = int(input("Ingrese la cantidad de dinero en euros: "))
13     if cantidad < 0:
14         print("La cantidad debe ser un número entero positivo.")
15     else:
16         b20, b10, b5, b1 = calcular_billetes(cantidad)
17         print("\nDistribución óptima de billetes:")
18         print(f"Billetes de 20€: {b20}")
19         print(f"Billetes de 10€: {b10}")
20         print(f"Billetes de 5€: {b5}")
21         print(f"Billetes de 1€: {b1}")
22     except ValueError:
23         print("Entrada inválida. Ingrese un número entero.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
```

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_19.py"

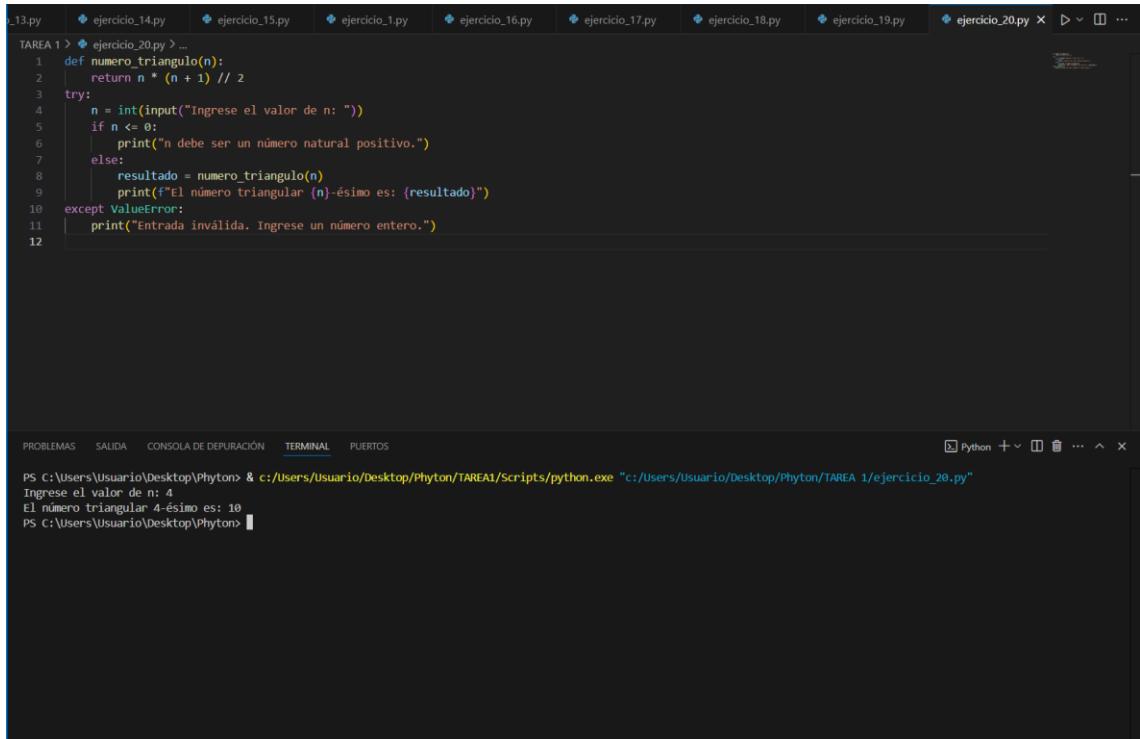
Ingrese la cantidad de dinero en euros: 45

Distribución óptima de billetes:

Billetes de 20€: 2
Billetes de 10€: 0
Billetes de 5€: 1
Billetes de 1€: 0

PS C:\Users\Usuario\Desktop\Python>

- Ejercicio 20



```

ejercicio_13.py ejercicio_14.py ejercicio_15.py ejercicio_16.py ejercicio_17.py ejercicio_18.py ejercicio_19.py ejercicio_20.py
TAREA 1 > ejercicio_20.py > ...
1 def numero_triangulo(n):
2     return n * (n + 1) // 2
3 try:
4     n = int(input("Ingrese el valor de n: "))
5     if n <= 0:
6         print("n debe ser un número natural positivo.")
7     else:
8         resultado = numero_triangulo(n)
9         print(f"El número triangular {n}-ésimo es: {resultado}")
10    except ValueError:
11        print("Entrada inválida. Ingrese un número entero.")
12

```

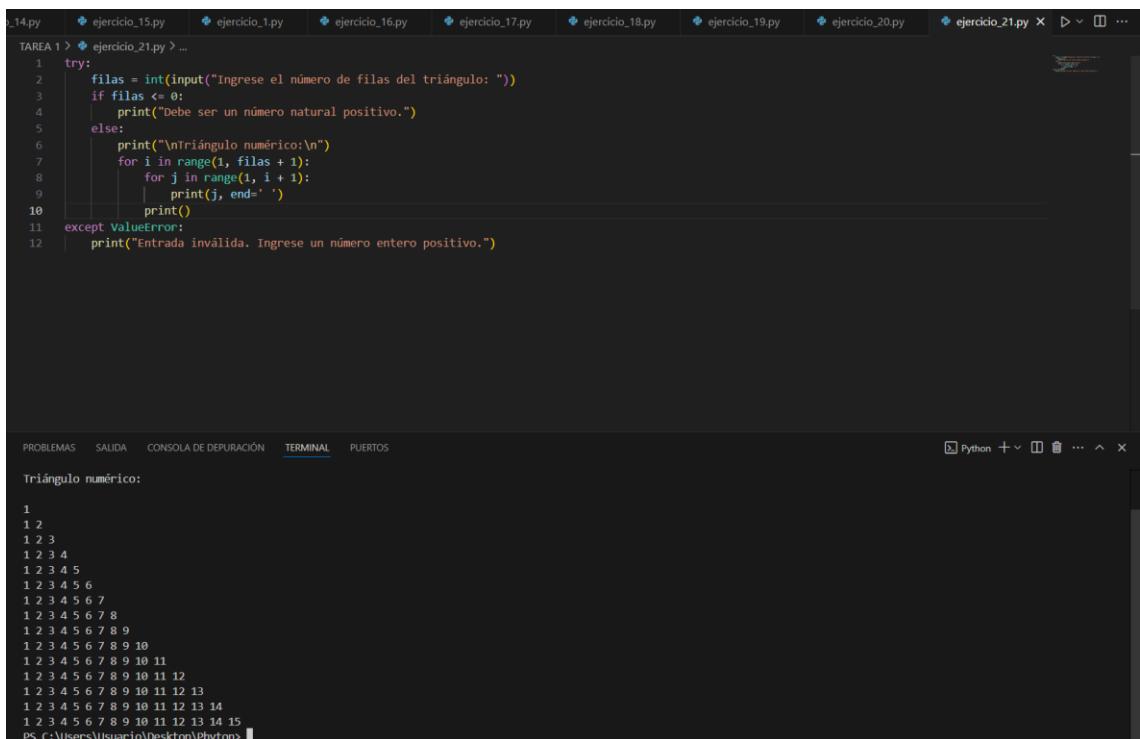
PROBLEMAS SALIDA CONSOLA DE DÉPURAÇÃO TERMINAL PUERTOS

```

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_20.py"
Ingrese el valor de n: 4
El número triangular 4-ésimo es: 10
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 21



```

ejercicio_14.py ejercicio_15.py ejercicio_16.py ejercicio_17.py ejercicio_18.py ejercicio_19.py ejercicio_20.py ejercicio_21.py
TAREA 1 > ejercicio_21.py > ...
1 try:
2     filas = int(input("Ingrese el número de filas del triángulo: "))
3     if filas <= 0:
4         print("Debe ser un número natural positivo.")
5     else:
6         print("\nTriángulo numérico:\n")
7         for i in range(1, filas + 1):
8             for j in range(1, i + 1):
9                 print(j, end=' ')
10            print()
11    except ValueError:
12        print("Entrada inválida. Ingrese un número entero positivo.")

```

PROBLEMAS SALIDA CONSOLA DE DÉPURAÇÃO TERMINAL PUERTOS

```

PS C:\Users\Usuario\Desktop\Python>

```

Triángulo numérico:

```

1
1 2
1 2 3
1 2 3 4
1 2 3 4 5
1 2 3 4 5 6
1 2 3 4 5 6 7
1 2 3 4 5 6 7 8
1 2 3 4 5 6 7 8 9
1 2 3 4 5 6 7 8 9 10
1 2 3 4 5 6 7 8 9 10 11
1 2 3 4 5 6 7 8 9 10 11 12
1 2 3 4 5 6 7 8 9 10 11 12 13
1 2 3 4 5 6 7 8 9 10 11 12 13 14
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 22

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
[101 rows x 3 columns]
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA1/ejercicio_22.py"
      N      N^2      N^0.5
0  1.000  1.000000  1.000000
1  1.001  1.002001  1.000500
2  1.002  1.004004  1.001000
3  1.003  1.006009  1.001499
4  1.004  1.008016  1.001998
..   ...
96 1.096  1.201216  1.046900
97 1.097  1.203409  1.047378
98 1.098  1.205604  1.047855
99 1.099  1.207801  1.048332
100 1.100  1.210000  1.048809
[101 rows x 3 columns]
PS C:\Users\Usuario\Desktop\Python>

```

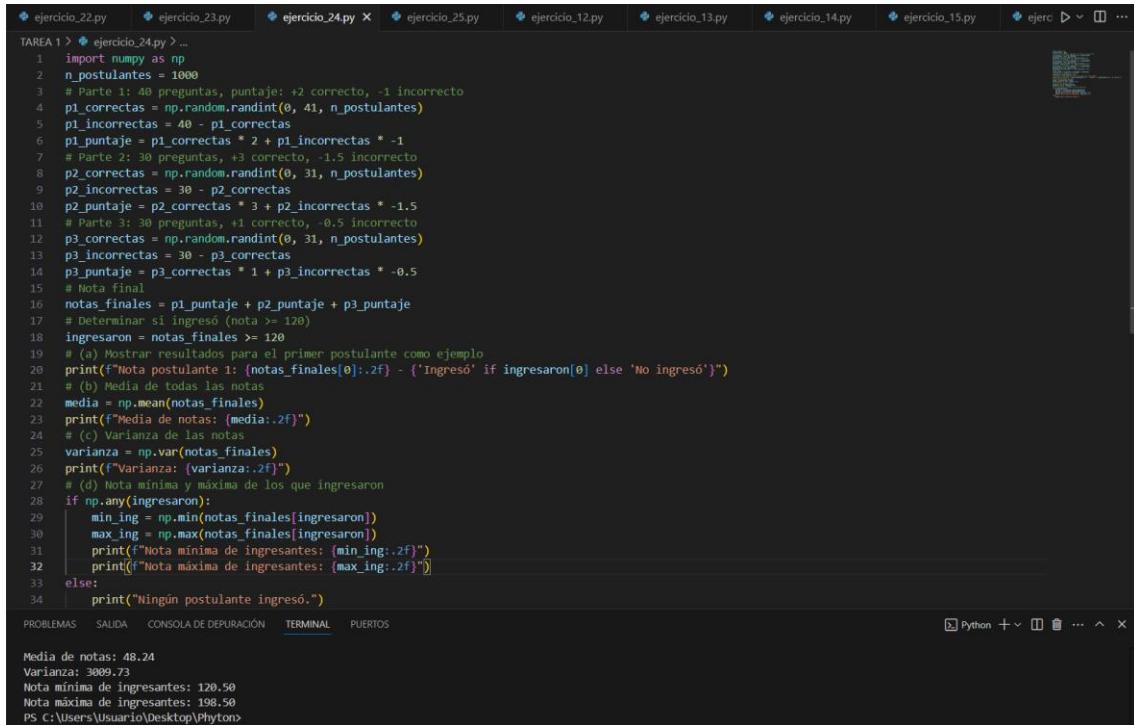
- Ejercicio 23

```

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
TAREA 1 > & ejercicio_23.py > multiplicar_por_11
1 def multiplicar_por_11(numero):
2     digitos = [int(d) for d in str(numero)]
3     resultado = []
4     resultado.append(digitos[0])
5     for i in range(1, len(digitos)):
6         suma = digitos[i] + digitos[i-1]
7         resultado.append(suma)
8     # Agrega el último dígito
9     resultado.append(digitos[-1])
10    # Manejo de acarreo
11    resultado = resultado[::-1] # invertimos para trabajar con unidades primero
12    for i in range(len(resultado)):
13        if resultado[i] >= 10:
14            carry = resultado[i] // 10
15            resultado[i] = resultado[i] % 10
16            if i + 1 < len(resultado):
17                resultado[i + 1] += carry
18            else:
19                resultado.append(carry)
20    resultado = resultado[::-1] # volvemos a invertir
21    return int("".join(str(d) for d in resultado))
22 #Por ejemplo
23 numero = 3247
24 print(f"{numero} x 11 = {multiplicar_por_11(numero)}")
PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA1/ejercicio_23.py"
3247 x 11 = 35717
PS C:\Users\Usuario\Desktop\Python>

```

- Ejercicio 24



```

ejercicio_22.py ejercicio_23.py ejercicio_24.py X ejercicio_25.py ejercicio_12.py ejercicio_13.py ejercicio_14.py ejercicio_15.py ejercicio D v ...
```

TAREA 1 > ejercicio_24.py > ...

```

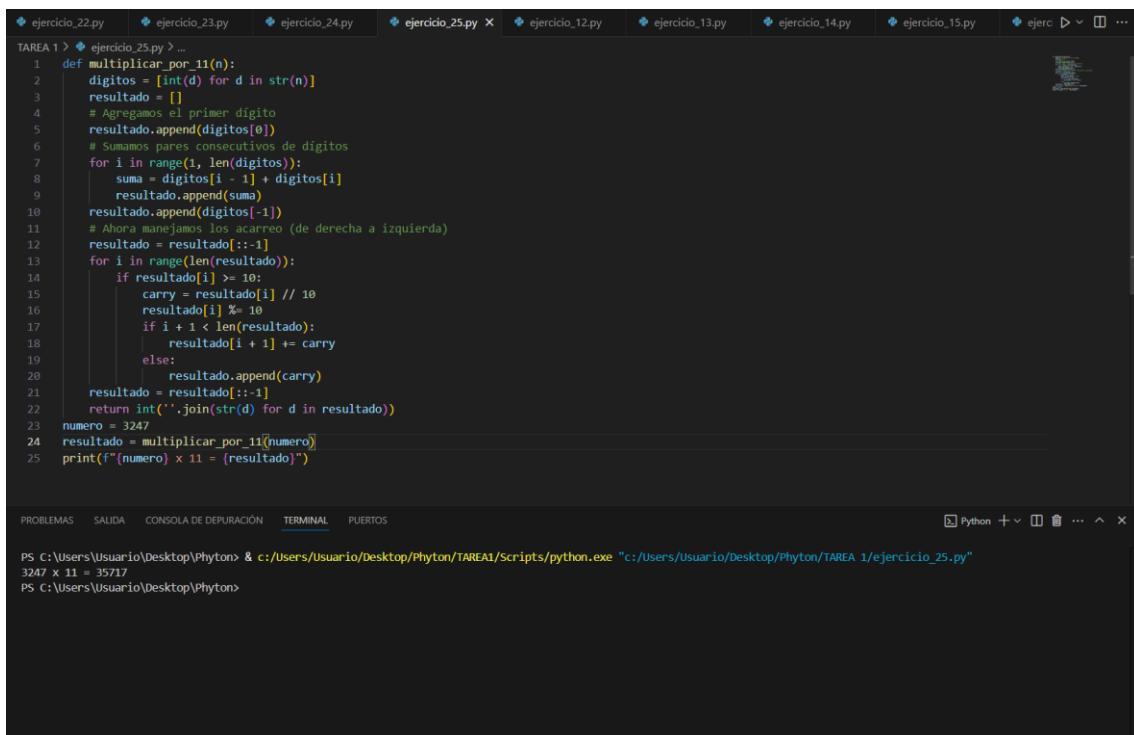
1 import numpy as np
2 n_postulantes = 1000
3 # Parte 1: 40 preguntas, puntaje: +2 correcto, -1 incorrecto
4 p1_correctas = np.random.randint(0, 41, n_postulantes)
5 p1_incorrectas = 40 - p1_correctas
6 p1_puntaje = p1_correctas * 2 + p1_incorrectas * -1
7 # Parte 2: 30 preguntas, +3 correcto, -0.5 incorrecto
8 p2_correctas = np.random.randint(0, 31, n_postulantes)
9 p2_incorrectas = 30 - p2_correctas
10 p2_puntaje = p2_correctas * 3 + p2_incorrectas * -0.5
11 # Parte 3: 30 preguntas, +1 correcto, -0.5 incorrecto
12 p3_correctas = np.random.randint(0, 31, n_postulantes)
13 p3_incorrectas = 30 - p3_correctas
14 p3_puntaje = p3_correctas * 1 + p3_incorrectas * -0.5
15 # Nota Final
16 notas_finales = p1_puntaje + p2_puntaje + p3_puntaje
17 # Determinar si ingresó (nota >= 120)
18 ingresaron = notas_finales >= 120
19 # (a) Mostrar resultados para el primer postulante como ejemplo
20 print(f"Nota postulante 1: {notas_finales[0]:.2f} - {'Ingresó' if ingresaron[0] else 'No ingresó'}")
21 # (b) Media de todas las notas
22 media = np.mean(notas_finales)
23 print(f"Media de notas: {media:.2f}")
24 # (c) Varianza de las notas
25 varianza = np.var(notas_finales)
26 print(f"Varianza: {varianza:.2f}")
27 # (d) Nota mínima y máxima de los que ingresaron
28 if np.any(ingresaron):
29     min_ing = np.min(notas_finales[ingresaron])
30     max_ing = np.max(notas_finales[ingresaron])
31     print(f"Nota mínima de ingresantes: {min_ing:.2f}")
32     print(f"Nota máxima de ingresantes: {max_ing:.2f}")
33 else:
34     print("Ningún postulante ingresó.")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
```

Media de notas: 48.24
Varianza: 3009.73
Nota mínima de ingresantes: 120.50
Nota máxima de ingresantes: 198.50

PS C:\Users\Usuario\Desktop\Python>

- Ejercicio 25



```

ejercicio_22.py ejercicio_23.py ejercicio_24.py ejercicio_25.py X ejercicio_12.py ejercicio_13.py ejercicio_14.py ejercicio_15.py ejercicio D v ...
```

TAREA 1 > ejercicio_25.py > ...

```

1 def multiplicar_por_11(n):
2     digitos = [int(d) for d in str(n)]
3     resultado = []
4     # Agregamos el primer dígito
5     resultado.append(digitos[0])
6     # Sumamos pares consecutivos de dígitos
7     for i in range(1, len(digitos)):
8         suma = digitos[i - 1] + digitos[i]
9         resultado.append(suma)
10    resultado.append(digitos[-1])
11    # Ahora manejamos los acarreos (de derecha a izquierda)
12    resultado = resultado[::-1]
13    for i in range(len(resultado)):
14        if resultado[i] >= 10:
15            carry = resultado[i] // 10
16            resultado[i] %= 10
17            if i + 1 < len(resultado):
18                resultado[i + 1] += carry
19            else:
20                resultado.append(carry)
21    resultado = resultado[::-1]
22    return int(''.join(str(d) for d in resultado))
23 numero = 3247
24 resultado = multiplicar_por_11(numero)
25 print(f"{numero} x 11 = {resultado}")

PROBLEMAS SALIDA CONSOLA DE DEPURACIÓN TERMINAL PUERTOS
```

PS C:\Users\Usuario\Desktop\Python> & c:/Users/Usuario/Desktop/Python/TAREA1/Scripts/python.exe "c:/Users/Usuario/Desktop/Python/TAREA 1/ejercicio_25.py"
3247 x 11 = 35717
PS C:\Users\Usuario\Desktop\Python>

EJERCICIOS PROPUESTOS (Dev- C++)

Elaborado por: Palacios Carrasco Roberto

ejercicio 1

The screenshot shows the Dev-C++ IDE interface. The project name is "ClasesFun" and the file is "tabla multiplicar del 3 y 9.cpp". The code prints multiplication tables for 3 and 9. The output window shows the results:

```

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 = 30

Tabla de multiplicar del 9:
9 x 1 = 9
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 = 90

```

Below the output, a message says "Process exited after 0.0914 seconds with return value 0 Presione una tecla para continuar . . .".

Ejercicio 2

The screenshot shows the Dev-C++ IDE interface. The project name is "ClasesFun" and the file is "ejercicio 2.cpp". The code reads a number from the user and calculates the sum of even numbers from 1 to the input, the sum of odd numbers from 1 to the input, and the sum of multiples of 5 from 1 to the input. The output window shows the results:

```

Introduce un numero entero: 10
Suma de pares desde 1 hasta 10: 30
Suma de impares desde 1 hasta 10: 25
Suma de multiplos de 5 desde 1 hasta 10: 15


```

Below the output, a message says "Process exited after 3.819 seconds with return value 0 Presione una tecla para continuar . . .".

Ejercicio 3

The screenshot shows the Dev-C++ IDE interface. The project name is "ClasesFun" and the file is "ejercicio 3.cpp". The code calculates the sum of the first 20 multiples of 3. The output window shows the result:

```

La suma de los 20 primeros multiplos de 3 es: 630


```

Below the output, a message says "Process exited after 0.08773 seconds with return value 0 Presione una tecla para continuar . . .".

Ejercicio 4

The screenshot shows the Dev-C++ IDE interface. On the left, the code for 'ejercicio 4.cpp' is displayed:

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     string numero;
7     cout << "Introduce un numero entero: ";
8     cin >> numero;
9
10    int suma = 0;
11    for (int i = 0; i < numero.length(); i++) {
12        suma += numero[i] - '0'; // Convertir char a int
13    }
14
15    int cantidad = numero.length();
16    int promedio = suma / cantidad; // División entera
17
18    // Realizar la rotación a la izquierda
19    for (int i = 0; i < cantidad - promedio; i++) {
20        char primero = numero[0];
21        numero.erase(0, 1);
22        numero.push_back(primerito);
23    }
24
25
26    cout << "Numero despues de rotar " << promedio << " veces: " << numero << endl;
27
28    return 0;
29 }

```

On the right, the terminal window shows the program's output:

```

Introduce un numero entero: 3452
Numero despues de rotar 3 veces: 2345
-----
Process exited after 28.06 seconds with return value
0
Presione una tecla para continuar . . .

```

At the bottom, the status bar indicates the compilation was successful:

```

Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 4.cpp
Output Size: 1,83309745788574 MiB
Compilation Time: 0,47s

```

Ejercicio 5

The screenshot shows the Dev-C++ IDE interface. On the left, the code for 'ejercicio 5.cpp' is displayed:

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 bool esBisiesto(int anio) {
6     return (anio % 4 == 0 && anio % 100 != 0) || (anio % 400 == 0);
7 }
8
9 int main() {
10    int anio;
11    int contadorTotal = 0;
12    int contadorBisiestos = 0;
13
14    while (contadorTotal < 10 && contadorBisiestos < 3) {
15        cout << "Introduce un año: ";
16        cin >> anio;
17
18        if (esBisiesto(anio)) {
19            cout << anio << " es bisiesto." << endl;
20            contadorBisiestos++;
21        } else {
22            cout << anio << " no es bisiesto." << endl;
23        }
24
25        contadorTotal++;
26
27    cout << "\nFin del programa. Se leyeron 5 años, de los cuales ";
28    cout << contadorTotal << " años, de los cuales " << contadorBisiestos << " fueron bisiestos." << endl;
29
30    return 0;
31 }

```

On the right, the terminal window shows the program's output:

```

Introduce un año: 2000
2000 es bisiesto.
Introduce un año: 2012
2012 es bisiesto.
Introduce un año: 1939
1939 no es bisiesto.
Introduce un año: 1838
1838 no es bisiesto.
Introduce un año: 2028
2028 es bisiesto.
Fin del programa. Se leyeron 5 años, de los cuales
3 fueron bisiestos.
-----
Process exited after 70.35 seconds with return value
0
Presione una tecla para continuar . . .

```

At the bottom, the status bar indicates the compilation was successful:

```

Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 5.cpp
Output Size: 1,83314037322998 MiB
Compilation Time: 0,47s

```

Ejercicio 6

The screenshot shows the Dev-C++ IDE interface. On the left, the code for 'ejercicio 6.cpp' is displayed:

```

1 #include <iostream>
2 #include <cmath>
3 using namespace std;
4
5 int main() {
6     double grados;
7
8     cout << "Introduce un angulo en grados (0 a 360): ";
9     cin >> grados;
10
11    if (grados < 0 || grados > 360) {
12        cout << "Error: El ángulo debe estar entre 0 y 360 grados." << endl;
13    } else {
14        // Convertir grados a radianes
15        double radianes = grados * (M_PI / 180.0);
16        double seno = sin(radianes);
17
18        cout << "El seno aproximado de " << grados << " grados es: " << seno << endl;
19
20    }
21
22    return 0;
23 }

```

On the right, the terminal window shows the program's output:

```

Introduce un angulo en grados (0 a 360): 30
El seno aproximado de 30 grados es: 0.5
-----
Process exited after 13.35 seconds with return value
0
Presione una tecla para continuar . . .

```

At the bottom, the status bar indicates the compilation was successful:

```

Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 6.cpp
Output Size: 1,83693504333496 MiB
Compilation Time: 0,45s

```

Ejercicio 7

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 7.cpp - [Executing] - Dev-C++ 5.11

Introduce un numero: 50
El numero es mayor.
Introduce un numero: 80
El numero es menor.
Introduce un numero: 60
El numero es mayor.
Introduce un numero: 70
El numero es menor.
Introduce un numero: 68
El numero es menor.
Introduce un numero: 66
El numero es menor.
Introduce un numero: 65
El numero es menor.
Introduce un numero: 63
El numero es mayor.
Introduce un numero: 64
¡Aciertaste!!
Adivinaste el número en 9 intento(s.).

Process exited after 72.49 seconds with return value 0
Presione una tecla para continuar . . .

```

Compilador Recursos Registro de Compilación Depuración Resultados Cerrar
 Cancelar Compilación
 - Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 7.cpp
 - Output Size: 1,0335933653027 MiB
 - Compilation Time: 0,50s
 Shorten compiler paths

Line: 28 Col: 27 Sel: 0 Lines: 36 Length: 992 Insertar Done parsing in 0,125 seconds

Ejercicio 8

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 8.cpp - [Executing] - Dev-C++ 5.11

Introduzca un numero: 6
La tabla de multiplicar del 6 es:
6 x 1 = 6
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 = 60

Process exited after 77.84 seconds with return value 0
Presione una tecla para continuar . . .

```

Compilador Recursos Registro de Compilación Depuración Resultados Cerrar
 Cancelar Compilación
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 8.cpp
 - Output Size: 1,83279037475586 MiB
 - Compilation Time: 1,20s
 Shorten compiler paths

Line: 10 Col: 30 Sel: 0 Lines: 25 Length: 603 Insertar Done parsing in 0,781 seconds

Ejercicio 9

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 9.cpp - [Executing] - Dev-C++ 5.11

Ingrese el valor inicial n: 1
Ingrese el valor final m: 10
Serie de Fibonacci entre 1 y 10:
1 2 3 5 8

Process exited after 25.93 seconds with return value 0
Presione una tecla para continuar . . .

```

Compilador Recursos Registro de Compilación Depuración Resultados Cerrar
 Cancelar Compilación
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 9.cpp
 - Output Size: 1,83279037475586 MiB
 - Compilation Time: 1,20s
 Shorten compiler paths

Line: 38 Col: 1 Sel: 0 Lines: 38 Length: 725 Insertar Done parsing in 0 seconds

Ejercicio 10

The screenshot shows the Dev-C++ IDE interface. On the left, the code for `ejercicio 10.cpp` is displayed:

```

1 //include <iomanip> // Para reverse()
2
3 using namespace std;
4
5 // Función que convierte un grupo de 4 bits en un carácter hexadecimal
6 char binarioAdecimal(string grupoBits) {
7     if (grupoBits == "0000") return '0';
8     if (grupoBits == "0001") return '1';
9     if (grupoBits == "0010") return '2';
10    if (grupoBits == "0011") return '3';
11    if (grupoBits == "0100") return '4';
12    if (grupoBits == "0101") return '5';
13    if (grupoBits == "0110") return '6';
14    if (grupoBits == "0111") return '7';
15    if (grupoBits == "1000") return '8';
16    if (grupoBits == "1001") return '9';
17    if (grupoBits == "1010") return 'A';
18    if (grupoBits == "1011") return 'B';
19    if (grupoBits == "1100") return 'C';
20    if (grupoBits == "1101") return 'D';
21    if (grupoBits == "1110") return 'E';
22    if (grupoBits == "1111") return 'F';
23    return ' ';
24 } // Cada bit es
25
26 int main() {
27     string binario;
28
29     cout << "Ingrese un numero binario desde el teclado ";
30     cin << "Ingresar un numero binario: ";
31     cin >> binario;
32
33     // Asignar que el numero tiene longitud múltiplo de 4
34     int longitud = binario.length();
35     int relleno = 4 - (longitud % 4);
36     if (relleno != 0) {
37         binario = string(relleno, '0') + binario; // agregar ceros a la izquierda
38     }
39
40     string hexadecimal = "";
41
42     // Convertir cada grupo de 4 bits en hexadecimal
43     for (size_t i = 0; i < binario.length(); i += 4) {
44         string grupo = binario.substr(i, 4);
45         hexadecimal += binarioAdecimal(grupo);
46     }
47
48     cout << "El numero hexadecimal es: " << hexadecimal << endl;
49
50     return 0;
51 }

```

On the right, the terminal window shows the program's output:

```

Ingrese un numero binario: 1010111101
El numero hexadecimal es: 2BD

Process exited after 22.66 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 11

The screenshot shows the Dev-C++ IDE interface. On the left, the code for `ejercicio 11.cpp` is displayed:

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     cout << "Ingrese el numero total de encuestados ";
7     cin >> n;
8
9     for (int i = 1; i <= n; i++) {
10        cout << "Encuestado " << i << " - ";
11        cout << "Acepta el producto A (1/0)? ";
12        cin >> a;
13        cout << "Acepta el producto B (1/0)? ";
14        cin >> b;
15
16        total++;
17
18        if (a == 1 && b == 1)
19            soloAB++;
20        else if (a == 0 && b == 1)
21            soloB++;
22        else if (a == 1 && b == 0)
23            ambos++;
24        else
25            ninguno++;
26    }
27
28    // Mostrar resultados
29    cout << fixed << setprecision(2); // para porcentajes con 2 decimales
30
31    cout << "Total de encuestados: " << total << endl;
32    cout << "100.0 * (soloA + ambos) / total << "% << endl;
33    cout << "Porcentaje que acepta el producto A: ";
34    cout << "100.0 * soloA / total << "% << endl;
35    cout << "Porcentaje que acepta el producto B: ";
36    cout << "100.0 * soloB / total << "% << endl;
37    cout << "Porcentaje que acepta ambos productos: ";
38    cout << "100.0 * ambos / total << "% << endl;
39    cout << "Porcentaje que acepta solo A: ";
40    cout << "33.33%" << endl;
41    cout << "Porcentaje que acepta solo B: ";
42    cout << "33.33%" << endl;
43    cout << "Porcentaje que no acepta ninguno: ";
44    cout << "0.00%" << endl;
45
46    return 0;
47 }

```

On the right, the terminal window shows the program's output:

```

Ingrese el numero total de encuestados: 3
Encuestado #1 - ✓Acepta el producto A (1/0)? 1
Encuestado #2 - ✓Acepta el producto B (1/0)? 0
Encuestado #3 - ✓Acepta el producto A (1/0)? 1
Encuestado #4 - ✓Acepta el producto B (1/0)? 1
Encuestado #5 - ✓Acepta el producto A (1/0)? 0
Encuestado #6 - ✓Acepta el producto B (1/0)? 1

Total de encuestados: 3
Porcentaje que acepta el producto A: 66.67%
Porcentaje que acepta el producto B: 66.67%
Porcentaje que acepta ambos productos: 33.33%
Porcentaje que acepta solo A: 33.33%
Porcentaje que acepta solo B: 33.33%
Porcentaje que no acepta ninguno: 0.00%

Process exited after 116.8 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 12

The screenshot shows the Dev-C++ IDE interface. On the left, the code for `ejercicio 12.cpp` is displayed:

```

1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int numero;
7
8     cout << "Ingrese un numero del 1 al 20: ";
9     cin >> numero;
10
11    if (numero < 1 || numero > 20) {
12        cout << "Numero fuera de rango (debe estar entre 1 y 20)." << endl;
13    } else if (numero == 1) {
14        cout << "UNO" << endl;
15    } else if (numero == 2) {
16        cout << "DOS" << endl;
17    } else if (numero == 3) {
18        cout << "TRES" << endl;
19    } else if (numero == 4) {
20        cout << "CUATRO" << endl;
21    } else if (numero == 5) {
22        cout << "CINCO" << endl;
23    } else if (numero == 6) {
24        cout << "SEIS" << endl;
25    } else if (numero == 7) {
26        cout << "Siete" << endl;
27    } else if (numero == 8) {
28        cout << "OCHO" << endl;
29    } else if (numero == 9) {
30        cout << "NUEVE" << endl;
31    } else if (numero == 10) {
32        cout << "DIEZ" << endl;
33    } else if (numero == 11) {
34        cout << "ONCE" << endl;
35    } else if (numero == 12) {
36        cout << "DOCE" << endl;
37    } else if (numero == 13) {
38        cout << "TRECE" << endl;
39    } else if (numero == 14) {
40        cout << "CATORCE" << endl;
41    } else if (numero == 15) {
42        cout << "QUINCE" << endl;
43    } else if (numero == 16) {
44        cout << "DIECISEIS" << endl;
45    } else if (numero == 17) {
46        cout << "DIECIOCHO" << endl;
47    } else if (numero == 18) {
48        cout << "DIECINUEVE" << endl;
49    } else if (numero == 19) {
50        cout << "VEINTE" << endl;
51    } else {
52        cout << "CERO" << endl;
53    }
54
55    return 0;
56 }

```

On the right, the terminal window shows the program's output:

```

Ingrese un numero del 1 al 20: 15
XV

Process exited after 13.24 seconds with return value 0
Presione una tecla para continuar . . .

```

Ejercicio 13

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 13.cpp - [Executing] - Dev-C++ 5.11
Suma de los numeros pares entre 20 y 400: 40110
Producto de los pares entre 20 y 40 : 1337982976
-----
Process exited after 0.07836 seconds with return value
0
Presione una tecla para continuar . . .

```

Ejercicio 14

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 14.cpp - [Executing] - Dev-C++ 5.11
Ingrese numeros positivos (0 para terminar):
15
2
3
6
0
Menor numero: 2
Mayor numero: 15
Promedio: 6.5
-----
Process exited after 49.72 seconds with return value
0
Presione una tecla para continuar . . .

```

Ejercicio 15

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 15.cpp - [Executing] - Dev-C++ 5.11
Ingrese un numero entero positivo: 6
6 es un numero perfecto.
-----
Process exited after 9.003 seconds with return value
0
Presione una tecla para continuar . . .

```

Ejercicio 16

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 16.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 4.9.2 64-bit Release
Proyecto Clases/Fuente (globales)
gmon.out ejercicio 8.cpp ejercicio 9.cpp ejercicio 10.cpp ejercicio 11.cpp ejercicio 12.cpp ejercicio 13.cpp ejercicio 14.cpp ejercicio 15.cpp ejercicio 16.cpp

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, i, contador = 0;
6
7     cout << "Ingrese un numero entero positivo: ";
8     cin >> n;
9
10    if (n <= 1) {
11        cout << "El numero no es primo." << endl;
12        return 0;
13    }
14
15    for (i = 1; i <= n; i++) {
16        if (n % i == 0) {
17            contador++;
18        }
19    }
20
21    if (contador == 2) {
22        cout << n << " es un numero primo." << endl;
23    } else {
24        cout << n << " no es un numero primo." << endl;
25    }
26
27    return 0;
28}

```

Ingrese un numero entero positivo: 37
37 es un numero primo.

Process exited after 8.862 seconds with return value
0
Presione una tecla para continuar . . .

Ejercicio 17

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 17.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 4.9.2 64-bit Release
Proyecto Clases/Fuente (globales)
gmon.out ejercicio 8.cpp ejercicio 9.cpp ejercicio 10.cpp ejercicio 11.cpp ejercicio 12.cpp ejercicio 13.cpp ejercicio 14.cpp ejercicio 15.cpp ejercicio 16.cpp ejercicio 17.cpp

1 #include <iostream>
2 using namespace std;
3
4 const int NumEmpleados = 50;
5 const float FactorExtra = 1.5;
6
7 int main() {
8     float ptas_hora;
9     float horas_trabajadas;
10    float salario;
11    int contador = 1;
12
13    cout << "Introduce el pago por hora ordinaria: ";
14    cin >> ptas_hora;
15
16    while (contador <= NumEmpleados) {
17        cout << "\nEmpleado #" << contador << endl;
18        cout << "Introduce las horas trabajadas: ";
19        cin >> horas_trabajadas;
20
21        if (horas_trabajadas > 40) {
22            float horas_extra = horas_trabajadas - 40;
23            salario = (40 * ptas_hora) + (horas_extra * ptas_hora * FactorExtra);
24        } else {
25            salario = horas_trabajadas * ptas_hora;
26        }
27
28        cout << "Salario del empleado #" << contador << ":" << salario << " unidades monetarias." << endl;
29        contador++;
30    }
31
32    return 0;
33}

```

Introduce el pago por hora ordinaria: 80

Empleado #1
Introduce las horas trabajadas: 8
Salario del empleado #1: 640 unidades monetarias.

Empleado #2
Introduce las horas trabajadas: |

Ejercicio 18

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 18.cpp - [Executing] - Dev-C++ 5.11
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
TDM-GCC 4.9.2 64-bit Release
Proyecto Clases/Fuente (globales)
gmon.out ejercicio 8.cpp ejercicio 9.cpp ejercicio 10.cpp ejercicio 11.cpp ejercicio 12.cpp ejercicio 13.cpp ejercicio 14.cpp ejercicio 15.cpp ejercicio 16.cpp ejercicio 17.cpp ejercicio 18.cpp

1 #include <iostream>
2 #include <cmath> // Para user sqrt()
3 using namespace std;
4
5 int main() {
6     int P;
7
8     cout << "Introduce el valor de P (cantidad de numeros naturales): ";
9     cin >> P;
10
11    for (int i = 1; i <= P; i++) {
12        int cuadrado = i * i;
13        int cubo = i * i * i;
14        double raiz = sqrt(i); // Función predefinida para la raíz cuadrada
15
16        cout << "Número: " << i
17        << " Cuadrado: " << cuadrado
18        << " Cubo: " << cubo
19        << " Raiz cuadrada: " << raiz << endl;
20    }
21
22    return 0;
23}

```

Introduce el valor de P (cantidad de numeros naturales): 10

Número: 1 | Cuadrado: 1 | Cubo: 1 | Raiz cuadrada: 1
Número: 2 | Cuadrado: 4 | Cubo: 8 | Raiz cuadrada: 1.41421
Número: 3 | Cuadrado: 9 | Cubo: 27 | Raiz cuadrada: 1.73205
Número: 4 | Cuadrado: 16 | Cubo: 64 | Raiz cuadrada: 2
Número: 5 | Cuadrado: 25 | Cubo: 125 | Raiz cuadrada: 2.23607
Número: 6 | Cuadrado: 36 | Cubo: 216 | Raiz cuadrada: 2.44949
Número: 7 | Cuadrado: 49 | Cubo: 343 | Raiz cuadrada: 2.64575
Número: 8 | Cuadrado: 64 | Cubo: 512 | Raiz cuadrada: 2.82843
Número: 9 | Cuadrado: 81 | Cubo: 729 | Raiz cuadrada: 3
Número: 10 | Cuadrado: 100 | Cubo: 1000 | Raiz cuadrada: 3.16228

Ejercicio 19

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ejercicio 19.cpp'. The right pane shows the terminal window with the program's output.

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int cantidad;
6     int billetes20, billetes10, billetes5, monedas1;
7
8     cout << "Introduce la cantidad de dinero en euros: ";
9     cin >> cantidad;
10
11    billetes20 = cantidad / 20;
12    cantidad %= 20;
13
14    billetes10 = cantidad / 10;
15    cantidad %= 10;
16
17    billetes5 = cantidad / 5;
18    cantidad %= 5;
19
20    monedas1 = cantidad; // Lo que queda son monedas de 1 euro
21
22    cout << "Desglose optimo:" << endl;
23    cout << "Billetes de 20 euros: " << billetes20 << endl;
24    cout << "Billetes de 10 euros: " << billetes10 << endl;
25    cout << "Billetes de 5 euros: " << billetes5 << endl;
26    cout << "Monedas de 1 euro: " << monedas1 << endl;
27
28    return 0;
29 }

```

Introduce la cantidad de dinero en euros: 67
Desglose optimo:
Billetes de 20 euros: 3
Billetes de 10 euros: 0
Billetes de 5 euros: 1
Monedas de 1 euro: 2

Process exited after 12.1 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 20

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ejercicio 20.cpp'. The right pane shows the terminal window with the program's output.

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int n, numeroTriangular;
6
7     cout << "Introduce el valor de n: ";
8     cin >> n;
9
10    numeroTriangular = n * (n + 1) / 2;
11
12    cout << "El numero triangular n-ésimo es: " << numeroTriangular << endl;
13
14    return 0;
15 }

```

Introduce el valor de n: 8
El numero triangular n-ésimo es: 36

Process exited after 9.548 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 21

The screenshot shows the Dev-C++ IDE interface. The left pane displays the source code for 'ejercicio 21.cpp'. The right pane shows the terminal window with the program's output.

```

1 #include <iostream>
2 using namespace std;
3
4 int main() {
5     int filas, numero = 1;
6
7     cout << "Introduce el numero de filas para el triangulo de Floyd: ";
8     cin >> filas;
9
10    for (int i = 1; i <= filas; i++) {
11        for (int j = 1; j <= i; j++) {
12            cout << numero << " ";
13            numero++;
14        }
15        cout << endl;
16    }
17
18    return 0;
19 }

```

Introduce el numero de filas para el triangulo de Floyd: 10
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

Process exited after 24.28 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 22

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 22.cpp
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
ejercicio 22.cpp
1 #include <iostream>
2 #include <iomanip> // Para usar setprecision y setw
3 #include <cmath> // Para pow() y sqrt()
4
5 using namespace std;
6
7 int main() {
8     double N = 1.0;
9
10    cout << setw(10) << "N"
11    << setw(15) << "N^2"
12    << setw(15) << "N^0.5" << endl;
13
14    while (N <= 1.1) {
15        double N2 = N * N;
16        double N05 = sqrt(N); // Equivalente a pow(N, 0.5)
17
18        cout << fixed << setprecision(3);
19        cout << setw(10) << N;
20        cout << setw(15) << setprecision(6) << N2;
21        cout << setw(15) << setprecision(6) << N05 << endl;
22
23        N += 0.001;
24    }
25
26    return 0;
27 }

```

N	N^2	N^0.5
1.000	1.000000	1.000000
1.001	1.002001	1.000500
1.002	1.004004	1.001000
1.003	1.006009	1.001499
1.004	1.008016	1.001998
1.005	1.010025	1.002497
1.006	1.012036	1.002996
1.007	1.014049	1.003494
1.008	1.016064	1.003992
1.009	1.018081	1.004490
1.010	1.020100	1.004988
1.011	1.022121	1.005485
1.012	1.024144	1.005982
1.013	1.026169	1.006479
1.014	1.028196	1.006976
1.015	1.030225	1.007472
1.016	1.032256	1.007968
1.017	1.034289	1.008464
1.018	1.036324	1.008960
1.019	1.038361	1.009455
1.020	1.040400	1.009950
1.021	1.042441	1.010445
1.022	1.044484	1.010940
1.023	1.046529	1.011435
1.024	1.048576	1.011929
1.025	1.050625	1.012423
1.026	1.052676	1.012917
1.027	1.054729	1.013410
1.028	1.056784	1.013903
1.029	1.058841	1.014396
1.030	1.060900	1.014889

Ejercicio 23

```

C:\Users\Roberto\OneDrive - Universidad Nacional Mayor de San Marcos\Whiteboards\Escritorio\trabajos C++\ejercicios\ejercicio 23.cpp
Archivo Edición Buscar Ver Proyecto Ejecutar Herramientas AStyle Ventana Ayuda
ejercicio 22.cpp ejercicio 23.cpp
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 bool esDivisiblePor11(string numero) {
6     int sumaAltern = 0;
7
8     for (int i = 0; i < numero.length(); i++) {
9         int digito = numero[i] - '0';
10
11         // Alternar signos: + - + - ...
12         if (i % 2 == 0) {
13             sumaAltern += digito;
14         } else {
15             sumaAltern -= digito;
16         }
17     }
18
19     return (sumaAltern % 11 == 0);
20 }
21
22 int main() {
23     string numero;
24
25     cout << "Introduce un numero entero: ";
26     cin >> numero;
27
28     if (esDivisiblePor11(numero)) {
29         cout << "El numero " << numero << " ES divisible por 11." << endl;
30     } else {
31         cout << "El numero " << numero << " NO es divisible por 11." << endl;
32     }
33
34     return 0;
35 }
36

```

Introduce un numero entero: 4015
El numero 4015 ES divisible por 11.

Process exited after 25.61 seconds with return value 0
Presione una tecla para continuar . . .

Ejercicio 24

```

1 #include <iostream>
2 #include <vector>
3 #include <cstdlib>
4 #include <ctime>
5 #include <cmath>
6 #include <limits>
7
8 using namespace std;
9
10 const int NUM_POSTULANTES = 1000;
11 const int NOTA_MINIMA = 120;
12
13 struct Postulante {
14     int correctasRazonamiento, incorrectasRazonamiento;
15     int correctasCiencias, incorrectasCiencias;
16     int correctasLetras, incorrectasLetras;
17     double nota;
18     bool ingreso;
19 };
20
21 double calcularNota(Postulante p) {
22     double nota = 0;
23     nota += p.correctasRazonamiento * 2 - p.incorrectasRazonamiento * 1;
24     nota += p.correctasCiencias * 3 - p.incorrectasCiencias * 1.5;
25     nota += p.correctasLetras * 1 - p.incorrectasLetras * 0.5;
26     return nota;
27 }
28
29 int main() {
30     srand(time(0));
31     vector<Postulante> postulantes(NUM_POSTULANTES);
32     double sumaNotas = 0, sumaCuadrados = 0;
33     int totalIngresantes = 0;
34     double minNota = numeric_limits<double>::max();
35     double maxNota = numeric_limits<double>::min();
36
37     // Simulamos las respuestas de cada postulante (aleatorio)
38     for (int i = 0; i < NUM_POSTULANTES; i++) {
39         Postulante& p = postulantes[i];
40         // Se asegura que correctas + incorrectas no excedan el total de preguntas
41         p.correctasRazonamiento = rand() % 41;
42         p.incorrectasRazonamiento = 40 - p.correctasRazonamiento;
43
44         p.correctasCiencias = rand() % 31;
45         p.incorrectasCiencias = 30 - p.correctasCiencias;
46
47         p.correctasLetras = rand() % 31;
48         p.incorrectasLetras = 30 - p.correctasLetras;
49
50         p.nota = calcularNota(p);
51         p.ingreso = (p.nota >= NOTA_MINIMA);
52
53         sumaNotas += p.nota;
54         sumaCuadrados += p.nota * p.nota;
55
56         if (p.ingreso) {
57             totalIngresantes++;
58             if (p.nota < minNota) minNota = p.nota;
59             if (p.nota > maxNota) maxNota = p.nota;
60         }
61     }
62
63     double media = sumaNotas / NUM_POSTULANTES;
64     double varianza = (sumaCuadrados / NUM_POSTULANTES) - (media * media);
65
66     // Resultados
67     cout << "\nRESULTADOS GENERALES\n";
68     cout << "-----\n";
69     cout << "Media de las notas: " << media << endl;
70     cout << "Varianza de las notas: " << varianza << endl;
71     cout << "Total de ingresantes: " << totalIngresantes << endl;
72
73     if (totalIngresantes > 0) {
74         cout << "Nota minima de ingresantes: " << minNota << endl;
75         cout << "Nota maxima de ingresantes: " << maxNota << endl;
76     } else {
77         cout << "Ningun postulante ingreso." << endl;
78     }
79
80     return 0;
81 }
```

RESULTADOS GENERALES

Media de las notas: 46.631
 Varianza de las notas: 3146.73
 Total de ingresantes: 108
 Nota minima de ingresantes: 120.5
 Nota maxima de ingresantes: 191

Process exited after 0.08891 seconds with return value 0
 Presione una tecla para continuar . . . |

Ejercicio 25

```

1  #include <iostream>
2  #include <vector>
3  #include <string>
4  using namespace std;
5
6  string multiplicarPor11(string numero) {
7      int n = numero.size();
8      vector<int> resultado(n + 2, 0);
9
10     // Convertimos número a vector de dígitos con ceros al inicio
11     vector<int> digitos(n + 2, 0); // índice desde 0 a n+1
12     for (int i = 0; i < n; i++) {
13         digitos[i + 1] = numero[i] - '0';
14     }
15
16     // Sumar dígitos adyacentes
17     for (int i = 1; i <= n; i++) {
18         resultado[i] = digitos[i] + digitos[i - 1];
19     }
20
21     // Manejo de acarreo
22     for (int i = resultado.size() - 1; i >= 0; i--) {
23         if (resultado[i] >= 10) {
24             resultado[i - 1] += resultado[i] / 10;
25             resultado[i] %= 10;
26         }
27     }
28
29     // Convertir el resultado a string
30     string producto = "";
31     bool cerosIniciales = true;
32     for (int i = 0; i < resultado.size(); i++) {
33         int d = resultado[i];
34         if (cerosIniciales && d == 0) continue;
35         cerosIniciales = false;
36         producto += char(d + '0');
37     }
38
39     // Si todos eran ceros (caso extremo), devolvemos "0"
40     if (producto == "") return "0";
41
42     return producto;
43 }
44
45 int main() {
46     string numero;
47     cout << "Ingrese un número entero: ";
48     cin >> numero;
49
50     string resultado = multiplicarPor11(numero);
51
52     cout << "El producto de " << numero << " por 11 es: " << resultado << endl;
53
54     return 0;
55 }
```

Ingrese un número entero: 40
 El producto de 40 por 11 es: 440

Process exited after 24.95 seconds with return value 0
 Presione una tecla para continuar . . . |