

1. La función menor, toma tres enteros x, y y z, y devuelva el entero menor.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class EjercicioMenor { // Función para encontrar el menor de tres números public static int encontrarMenor(int x, int y, int z) { int menor = x; // Asumimos inicialmente que 'x' es el menor if (y < menor) { menor = y; // Si 'y' es menor, actualizamos 'menor' } if (z < menor) { menor = z; // Si 'z' es menor, actualizamos 'menor' } return menor; // Devolvemos el valor menor encontrado } public static void main(String[] args) { int num1, num2, num3; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num1: "); num1 = entrada.nextInt(); System.out.println("Introduzca el num2: "); num2 = entrada.nextInt(); System.out.println("Introduzca el num3: "); num3 = entrada.nextInt(); int resultado = encontrarMenor(num1, num2, num3); System.out.println("El menor de los tres números es: " + resultado); } } </pre>

Corrida:

```

run:
Introduzca el num1:
1
Introduzca el num2:
2
Introduzca el num3:
3
El menor de los tres números es: 1

```

2. Defina una función que devuelva un boolean, true si un número pasado como parámetro es primo y false si no lo es.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer2primo { public static boolean esPrimo(int n) { if (n <= 1) { return false; // 0 y 1 no son primos } int i = 2; int c = 0; // Contador de divisores while (i <= n) { if (n % i == 0) { c++; } i++; } if (c == 1) { return true; // Es primo } else { return false; // No es primo } } public static void main(String[] args) { int num; boolean resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num: "); num=entrada.nextInt(); resultado=esPrimo(num); // Modificación para imprimir "primo" o "no es primo" if (resultado) { System.out.println("El número " + num + " es primo"); } else { System.out.println("false"); } } } </pre>

Corrida:

```

run:
Introduzca el num:
3
El número 3 es primo

```

3. Defina una función que encuentre el enésimo primo

Análisis y diseño	Programa
	<pre>package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer3 { public static boolean esPrimo(int n) { if (n <= 1) { return false; // 0 y 1 no son primos } int i = 2; int c = 0; // Contador de divisores while (i <= n) { if (n % i == 0) { c++; } i++; } if (c == 1) { return true; // Es primo } else { return false; // No es primo } } public static int encontrarnPrimo(int n) { int numero = 2; // Comenzamos a buscar desde el primer primo int contadorPrimos = 0; while (contadorPrimos < n) { if (esPrimo(numero)) { contadorPrimos++; } numero++; } return numero - 1; // Restamos 1 porque incrementamos 'numero' una vez de más al final del bucle } public static void main(String[] args) { int num,resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num: "); num=entrada.nextInt();</pre>

```
        resultado=encontrarnPrimo(num);  
        System.out.println("El n-ésimo primo es:  
"+resultado);  
  
    }  
}
```

Corrida:

```
run:
```

```
Introduzca el num:
```

```
2
```

```
El n-ésimo primo es: 3
```

4. Defina una función que calcule el factorial de un número.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer4factorial { public static long calcularFactorial(int numero) { if (numero == 0 numero == 1) { return 1; // El factorial de 0 y 1 es 1 } else { long factorial = 1; for (int i = 2; i <= numero; i++) { factorial *= i; } return factorial; } } public static void main(String[] args) { int num; // Cambiamos 'num' a tipo 'int' long resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num: "); num = entrada.nextInt(); resultado = calcularFactorial(num); System.out.println("El factorial es: " + resultado); } } </pre>

Corrida:

```

run:
Introduzca el num:
3
El factorial es: 6

```

5. Defina un método con retorno que invierta un número.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer5inviertenum { public static int invertirNumero(int numero) { int numeroInvertido = 0; while (numero != 0) { int digito = numero % 10; // Obtener el último digito numeroInvertido = numeroInvertido * 10 + digito; // Agregar el dígito al número invertido numero /= 10; // Eliminar el último dígito del número original } return numeroInvertido; } public static void main(String[] args) { int num; // Cambiamos 'num' a tipo 'int' long resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num: "); num = entrada.nextInt(); resultado = invertirNumero(num); System.out.println("El num invertido es: " + resultado); } } </pre>

Corrida:

```

run:
Introduzca el num:
123
El num invertido es: 321

```

6. Defina un método con retorno que calcule la suma de los divisores de un número. Ejemplo Si n = 4 -> se devuelve 7 (1 + 2 + 4 =7)

Análisis y diseño	Programa
	<pre>package modular; import java.util.Scanner; public class Ejer6sumadivisores { public static int sumaDivisores(int numero) { int suma = 0; for (int i = 1; i <= numero; i++) { if (numero % i == 0) { // Si 'i' es un divisor suma += i; // Sumamos 'i' a la suma total } } return suma; } public static void main(String[] args) { int num; // Cambiamos 'num' a tipo 'int' long resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el num: "); num = entrada.nextInt(); resultado = sumaDivisores(num); System.out.println("La suma de divisores es: " + resultado); } }</pre>

Corrida:

```
run:
Introduzca el num:
4
La suma de divisores es: 7
```

7. Escriba un método con retorno que elimina el k-esimo dígito de un número

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; import static modular.Ejer6sumadivisores.sumaDivisores; public class Ejer7eliminakesimo { public static int eliminarDigito(int numero, int k) { int numeroSinDigito = 0; int posicion = 1; while (numero != 0) { int digito = numero % 10; if (posicion != k) { numeroSinDigito = numeroSinDigito * 10 + digito; } numero /= 10; posicion++; } // Invertir el número resultante para obtener el orden correcto int resultado = 0; while (numeroSinDigito != 0) { int digito = numeroSinDigito % 10; resultado = resultado * 10 + digito; numeroSinDigito /= 10; } return resultado; } public static void main(String[] args) { int num,k; // Cambiamos 'num' a tipo 'int' long resultado; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca el knum: "); k = entrada.nextInt(); System.out.println("Introduzca el num: "); num = entrada.nextInt(); resultado = eliminarDigito(num, k); System.out.println("El nuevo num es: " + resultado); } } </pre>

Corrida:

```

run:
Introduzca el knum:
2
Introduzca el num: |
1234
El nuevo num es: 124

```


8. (Números perfectos) Se dice que un número entero es un número perfecto si la suma de sus divisores, incluyendo 1 (pero no el número en sí), es igual al número. Por ejemplo, 6 es un número perfecto ya que $6 = 1 + 2 + 3$. Escriba una función llamada perfecto que determine si el parámetro número es un número perfecto. Use esta función en un programa que determine e imprima todos los números perfectos entre 1 y 1000. Imprima los divisores de cada número perfecto para confirmar que el número sea realmente perfecto. Ponga a prueba el poder de su computadora, evaluando números mucho más grandes que 1000.

Análisis y diseño	Programa
	<pre> package modular; public class Ejer8numperfecto { // Función para calcular la suma de los divisores de // un número (excluyendo el propio número) public static int sumaDivisores(int numero) { int suma = 0; for (int i = 1; i < numero; i++) { // Iteramos hasta 'numero - 1' if (numero % i == 0) { suma += i; } } return suma; } // Función para determinar si un número es // perfecto public static boolean esPerfecto(int numero) { return sumaDivisores(numero) == numero; } public static void main(String[] args) { System.out.println("Números perfectos entre 1 y 1000:"); for (int numero = 1; numero <= 1000; numero++) { if (esPerfecto(numero)) { System.out.print(numero + " = "); // Imprimir los divisores for (int i = 1; i < numero; i++) { if (numero % i == 0) { System.out.print(i + " "); } } System.out.println(); } } } } </pre>

Corrida:

```

run:
Números perfectos entre 1 y 1000:
6 = 1 2 3
28 = 1 2 4 7 14
496 = 1 2 4 8 16 31 62 124 248

```

9. Dado un número N verificar si es par o impar sin utilizar DIV, MOD, /, o la función parte entera o []

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer9parimparsindivmodent { public static boolean esPar(int numero) { while (numero > 1) { numero -= 2; } return numero == 0; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese un número entero: "); int numero = scanner.nextInt(); if (esPar(numero)) { System.out.println(numero + " es par."); } else { System.out.println(numero + " es impar."); } scanner.close(); // Cerrar el Scanner para liberar recursos } } </pre>

Corrida:

```

run:
Ingrese un número entero: 5
5 es impar.

```

10. Dado un número NUM entero positivo reemplazar los dígitos impares que tuviera por el dígito par superior
(en el caso del 9 reemplazar por 0).

Ejemplo:	NUM = 29652148
Entrada	NNUM = 20662248
Salida	

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer10reemplimparporparsuperior { public static int reemplazarImparesPorPares(int num) { int nuevoNum = 0; int multiplicador = 1; while (num > 0) { int digito = num % 10; if (digito % 2 != 0) { // Si es impar digito = (digito == 9) ? 0 : digito + 1; } nuevoNum += digito * multiplicador; multiplicador *= 10; num /= 10; } return nuevoNum; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); int resultado; System.out.print("Ingrese un número entero: "); int numero = scanner.nextInt(); resultado=reemplazarImparesPorPares(numero); System.out.println("Número original: " + numero); System.out.println("Número con impares reemplazados: " + resultado); } } </pre>

Corrida:

```

run:
Ingrese un número entero: 12345
Número original: 12345
Número con impares reemplazados: 22446

```

11. (Elimine Dígito) Leer un número Z y eliminar todos sus dígitos cuyo valor sea igual a otro número A previamente leído.

Ejemplo:

Entrada	Z = 47647684 y A = 4
Salida	Z = 76768

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer11eliminarDigitoA { public static int eliminarDigito(int numero, int digitoAEliminar) { int nuevoNumero = 0; int multiplicador = 1; while (numero > 0) { int digitoActual = numero % 10; if (digitoActual != digitoAEliminar) { nuevoNumero += digitoActual * multiplicador; multiplicador *= 10; } numero /= 10; } return nuevoNumero; } public static void main(String[] args) { int Z; int A; Scanner entrada = new Scanner(System.in); System.out.println("Introduzca Z"); Z=entrada.nextInt(); System.out.println("Introduzca A"); A=entrada.nextInt(); int resultado; resultado = eliminarDigito(Z, A); System.out.println("Número original (Z): " + Z); System.out.println("Dígito a eliminar (A): " + A); System.out.println("Número resultante: " + resultado); } } </pre>

Corrida:

```

47647684
Introduzca A
4
Número original (Z): 47647684
Dígito a eliminar (A): 4
Número resultante: 76768

```

12. La función menor, toma tres enteros x, y y z, y devuelva el entero menor.

Análisis y diseño	Programa
	<pre> package modular; public class Ejer12anadirdigcapicua { public static int convertirEnCapicua(int numero) { int copia = numero; // Copiamos el número original int reverso = 0; // Inicializamos el reverso int nDigitos = 0; // Contador de dígitos // Construimos el reverso del número while (copia > 0) { int digito = copia % 10; // Extraemos el último dígito reverso = reverso * 10 + digito; // Lo agregamos al reverso copia /= 10; // Eliminamos el último dígito de la copia nDigitos++; // Incrementamos el contador de digitos } // Aquí obtenemos los dígitos que se agregan int digitosAgregados = reverso % (int) Math.pow(10, nDigitos - 1); // Quitamos el último dígito del reverso // Formamos el número capicúa int capicua = numero * (int) Math.pow(10, nDigitos - 1) + digitosAgregados; // Mostramos los dígitos agregados System.out.println("Dígitos agregados: " + digitosAgregados); return capicua; } // Método main para probar la función public static void main(String[] args) { int X = 4568; // Cambia este valor para probar con otros números int R = convertirEnCapicua(X); System.out.println("El número capicúa es: " + R); } } </pre>

Corrida:

```

run:
Dígitos agregados: 654
El número capicúa es: 4568654

```

13. Dado un número entero Z, se pide ordenar ascendentemente los dígitos del número.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer132 { // Función para calcular la longitud de un número entero public static int calcularLongitud(int numero) { if (numero == 0) { return 1; } int longitud = 0; while (numero != 0) { numero /= 10; longitud++; } return longitud; } // Función auxiliar para verificar si un dígito está presente en un número public static boolean estaEnNumero(int numero, int digito) { while (numero > 0) { if (numero % 10 == digito) { return true; } numero /= 10; } return false; } // Función para obtener un dígito específico de un número public static int obtenerDigito(int numero, int posicion) { int divisor = (int) Math.pow(10, posicion); return (numero / divisor) % 10; } public static int ordenarDigitosDescendente(int numero) { int longitud = calcularLongitud(numero); int numeroOrdenado = 0; int multiplicador = 1; // Iteramos para cada posición del número ordenado for (int posicion = 0; posicion < longitud; posicion++) { int digitoMayor = 0; // Inicializamos con el valor mínimo posible para un dígito // Buscamos el dígito mayor que aún no hemos agregado al número ordenado </pre>

```
        for (int i = 0; i < longitud; i++) {
            int digitoActual =
obtenerDigito(numero, i);
            if (digitoActual > digitoMayor &&
!estaEnNumero(numeroOrdenado, digitoActual)) {
                digitoMayor = digitoActual;
            }
        }

        // Agregamos el dígito mayor al número
ordenado
        numeroOrdenado += digitoMayor *
multiplicador;
        multiplicador *= 10;
    }

    return numeroOrdenado;
}

// ... (la función estaEnNumero permanece igual)

public static void main(String[] args) {
    int Z,resultado;
    Scanner entrada=new Scanner(System.in);
    System.out.println("Introduzca Z");
    Z=entrada.nextInt();
    resultado = ordenarDigitosDescendente(Z);

    System.out.println("Ejemplo:");
    System.out.println("Número original (Z): " +
Z);

    System.out.println("Número con dígitos
ordenados ascendentemente: " + resultado);
}
}
```

Corrida:

```
run:
```

```
12546845
```

```
Ejemplo:
```

```
Número original (Z): 12546845
```

```
Número con dígitos ordenados ascendentemente: 124568
```

14. Leer 2 números A y B enteros y positivos cuyos dígitos están ordenados ascendentemente, obtener un tercer número X entero y positivo formado por los dígitos A y B, que también deberá estar ordenado ascendentemente

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; public class Ejer14nuevonum1234 { public static int fusionarNumerosDescendente(int A, int B) { int resultado = 0; int multiplicador = 1; while (A > 0 B > 0) { int digitoA; int digitoB; // Reemplazo del operador ternario para digitoA if (A > 0) { digitoA = A % 10; } else { digitoA = Integer.MIN_VALUE; } // Reemplazo del operador ternario para digitoB if (B > 0) { digitoB = B % 10; } else { digitoB = Integer.MIN_VALUE; } if (digitoA >= digitoB) { resultado += digitoA * multiplicador; A /= 10; } else { resultado += digitoB * multiplicador; B /= 10; } multiplicador *= 10; } return resultado; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese el primer número (A): "); int A = scanner.nextInt(); System.out.print("Ingrese el segundo número (B): "); int B = scanner.nextInt(); </pre>


```

        int X = fusionarNumerosDescendente(A, B);

        System.out.println("El número resultante (X)
es: " + X);

        scanner.close();
    }
}

```

Corrida:

```

run:
Ingrese el primer número (A): 12
Ingrese el segundo número (B): 34
El número resultante (X) es: 1234

```

15. Leer un número X entero y positivo y eliminar aquellos dígitos cuyo valor sea igual al menor de ellos.
Ejemplo: Entra X = 6753639 y A = 3
Sale NX = 67569

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer15eliminarDigmenor { public static int eliminarDigitosMinimos(int numero) { int digitoMinimo = 9; // Inicializamos con el valor máximo posible para un dígito int numeroAuxiliar = numero; // Copia para no modificar el original // Encontrar el dígito mínimo while (numeroAuxiliar > 0) { int digito = numeroAuxiliar % 10; if (digito < digitoMinimo) { digitoMinimo = digito; } numeroAuxiliar /= 10; } int nuevoNumero = 0; int multiplicador = 1; // Construir el nuevo número sin los dígitos mínimos while (numero > 0) { int digito = numero % 10; if (digito != digitoMinimo) { nuevoNumero += digito * multiplicador; multiplicador *= 10; } numero /= 10; } } } </pre>

```

        return nuevoNumero;
    }

    public static void main(String[] args) {
        int X,resultado;
        Scanner entrada=new Scanner(System.in);
        System.out.println("Introduzca el dig X");
        X=entrada.nextInt();
        resultado = eliminarDigitosMinimos(X);

        System.out.println("Ejemplo:");
        System.out.println("Número original (X): " +
X);
        System.out.println("Número resultante: " +
resultado);
    }
}

```

Corrida:

```

run:
Introduzca el dig X
1234
Ejemplo:
Número original (X): 1234
Número resultante: 234

```

16. Leer un número X en base 10 y convertirlo a base B, donde B>1 y B<10

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer16x10xB { public static int convertirABase(int numeroDecimal, int nuevaBase) { if (nuevaBase <= 1 nuevaBase >= 10) { throw new IllegalArgumentException("La nueva base debe estar entre 2 y 9."); } if (numeroDecimal == 0) { return 0; // Caso especial: 0 en cualquier base es 0 } int representacion = 0; int multiplicador = 1; while (numeroDecimal > 0) { int residuo = numeroDecimal % nuevaBase; representacion += residuo * multiplicador; </pre>

```

        multiplicador *= 10; // Desplazamos los
        dígitos existentes a la izquierda
        numeroDecimal /= nuevaBase;
    }

    return representacion;
}

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

    System.out.print("Ingrese el número decimal
(X): ");
    int X = entrada.nextInt();

    System.out.print("Ingrese la nueva base (B):
");
    int B = entrada.nextInt();

    int resultado = convertirABase(X, B);
    System.out.println("La representación de "
+ X + " en base " + B + " es: " + resultado);

}
}

```

Corrida:

```

run:
Ingrese el número decimal (X): 5
Ingrese la nueva base (B): 2
La representación de 5 en base 2 es: 101

```

17. El máximo común divisor (MCD) de dos enteros es el entero más grande que puede dividir uniformemente a cada uno de los dos números. Escriba una función llamada mcd que devuelva el máximo común divisor de dos enteros.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer17mcd { public static int mcd(int a, int b) { // Asegurarnos de que 'a' sea mayor o igual que 'b' if (b > a) { int temp = a; a = b; b = temp; } } } </pre>

```

// Algoritmo de Euclides
while (b != 0) {
    int temp = b;
    b = a % b;
    a = temp;
}

return a; // El MCD es el valor final de 'a'
}

public static void main(String[] args) {
    int numero1 ;
    int numero2;
    Scanner entrada=new Scanner(System.in);
    System.out.println("Introduzca el num1");
    numero1=entrada.nextInt();
    System.out.println("Introduzca el num2");
    numero2=entrada.nextInt();
    int resultado = mcd(numero1, numero2);

    System.out.println("El MCD de " + numero1 + "
y " + numero2 + " es: " + resultado);
}
}

```

Corrida:

```

run:
Introduzca el num1
48
Introduzca el num2
16
El MCD de 48 y 16 es: 16

```

18. Dado un número X en base 10 convertir a otra base B (2, 3, 4, 5, 6, 7, 8,9). Desplegar el número y su equivalente en base B

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer18x10xB { public static int convertirABase(int numeroDecimal, int nuevaBase) { if (numeroDecimal == 0) { return 0; } int representacion = 0; int multiplicador = 1; while (numeroDecimal > 0) { int residuo = numeroDecimal % nuevaBase; // Validación del dígito (sin excepciones) if (residuo < 0 residuo >= nuevaBase) { System.out.println("Error: Dígito inválido para la base especificada."); return -1; // Indicamos un error con un valor de retorno especial } </pre>

```
representacion += residuo * multiplicador;
multiplicador *= 10;
numeroDecimal /= nuevaBase;
}

return representacion;
}

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

    // Validación de la base (sin excepciones)
    do {
        System.out.print("Ingrese la nueva base (B, entre 2 y 9): ");
        B = scanner.nextInt();

        if (B <= 1 || B >= 10) {
            System.out.println("Error: La base debe estar entre 2 y 9.");
        }
    } while (B <= 1 || B >= 10);

    System.out.print("Ingrese el número decimal (X): ");
    int X = scanner.nextInt();

    int resultado = convertirABase(X, B);

    // Verificar si hubo un error en la conversión
    if (resultado == -1) {
        System.out.println("La conversión no se pudo realizar debido a un dígito inválido.");
    } else {
        System.out.println("La representación de " + X + " en base " + B + " es: " + resultado);
    }
}
```

Corrida:

```
run:
Ingrese la nueva base (B, entre 2 y 9): 10
Error: La base debe estar entre 2 y 9.
Ingrese la nueva base (B, entre 2 y 9): 5
Ingrese el número decimal (X): 5
La representación de 5 en base 5 es: 10
```

19. La función menor, toma tres enteros x, y y z, y devuelva el entero menor.

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer19xBX10 { public static int convertirADecimal(int numeroEnBaseB, int baseOriginal) { if (baseOriginal <= 1 baseOriginal >= 10) { System.out.println("Error: La base original debe estar entre 2 y 9."); return -1; // Indicamos un error con un valor de retorno especial } int numeroDecimal = 0; int potencia = 0; while (numeroEnBaseB > 0) { int digito = numeroEnBaseB % 10; if (digito < 0 digito >= baseOriginal) { System.out.println("Error: Dígito inválido para la base especificada."); return -1; // Indicamos un error } numeroDecimal += digito * calcularPotencia(baseOriginal, potencia); potencia++; numeroEnBaseB /= 10; } return numeroDecimal; } public static int calcularPotencia(int base, int exponente) { int resultado = 1; for (int i = 0; i < exponente; i++) { resultado *= base; } return resultado; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); int B; do { System.out.print("Ingresa la base original B (entre 2 y 9): "); B = scanner.nextInt(); if (B <= 1 B >= 10) { System.out.println("Error: La base debe estar entre 2 y 9."); } } while (B <= 1 B >= 10); System.out.print("Ingresa el número en base B: "); int X = scanner.nextInt(); int resultado = convertirADecimal(X, B); if (resultado == -1) { System.out.println("La conversión no se pudo realizar debido a un error en la entrada."); } else { System.out.println("El número " + X + " en base " + B + " es equivalente a " + resultado + " en base 10."); } } </pre>

```
}
}
```

Corrida:

```
run:
Ingrese la base original B (entre 2 y 9): 5
Ingrese el número en base B: 3
El número 3 en base 5 es equivalente a 3 en base 10.
```

20. Realizar las siguientes sumatorias para N términos, utilizando las funciones o procedimientos necesarios.

a.
$$S = \frac{5^2}{3!} + \frac{7^4}{5!} + \frac{9^6}{7!} + \frac{11^8}{9!} + \dots$$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20a { // Función para calcular el factorial de un número public static long factorial(int n) { if (n == 0) { return 1; } else { return n * factorial(n - 1); } } // Función para calcular la potencia de un número public static double potencia(double base, int exponente) { double resultado = 1; for (int i = 0; i < exponente; i++) { resultado *= base; } return resultado; } // Función para calcular la suma de la serie public static double calcularSumaSerie(int numTerminos) { double suma = 0; int numerador = 5; int denominadorFactorial = 3; for (int i = 0; i < numTerminos; i++) { double termino = potencia(numerador, 2 * i + 2) / factorial(denominadorFactorial); suma += termino; numerador += 2; denominadorFactorial += 2; } return suma; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese la cantidad de términos a sumar: "); int numTerminos = scanner.nextInt(); double resultado = calcularSumaSerie(numTerminos); System.out.println("El resultado de la suma de la serie es: " + resultado); } </pre>

```
}
}
```

Corrida:

```
run:
Ingrese la cantidad de términos a sumar: 2
El resultado de la suma de la serie es: 24.175
```

b. $S=1+2+3+5+7+11+13+.....$

Análisis y diseño	Programa
	<pre>package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20b { public static boolean esPrimo(int numero) { if (numero <= 1) { return false; } for (int i = 2; i <= Math.sqrt(numero); i++) { if (numero % i == 0) { return false; } } return true; } public static int calcularSumaNMenosUnoPrimos(int n) { int suma = 0; int contador = 0; int numero = 2; // Empezamos desde el primer primo while (contador < n - 1) { // Sumamos n-1 números primos if (esPrimo(numero)) { suma += numero; contador++; } numero++; } return suma + 1; // Sumamos 1 al final } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese la cantidad de números primos (n): "); int n = scanner.nextInt(); int resultado = calcularSumaNMenosUnoPrimos(n); System.out.println("La suma de los primeros " + (n - 1) + " números primos + 1 es: " + resultado); scanner.close(); } }</pre>

```
run:
Ingrese la cantidad de números primos (n): 4
La suma de los primeros 3 números primos + 1 es: 11
```

Corrida

c. $S = 1^0 + 2^1 + 3^2 + 4^3 + 5^4 + 6^5 + \dots$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20c { // Función para calcular la potencia de un número public static double potencia(double base, int exponente) { return Math.pow(base, exponente); } // Función para calcular la suma de la serie public static double calcularSumaSerie(int numTerminos) { double suma = 0; for (int i = 1; i <= numTerminos; i++) { double termino = potencia(i, i - 1); // Elevamos 'i' a la potencia 'i - 1' suma += termino; } return suma; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese la cantidad de términos a sumar: "); int numTerminos = scanner.nextInt(); double resultado = calcularSumaSerie(numTerminos); System.out.println("El resultado de la suma de la serie es: " + resultado); scanner.close(); } } </pre>

Corrida:

run:

Ingrese la cantidad de términos a sumar: 3

El resultado de la suma de la serie es: 12.0

d. $S = 2^3 + 3^4 + 4^5 + 5^6 + 6^7 + 7^8 + \dots$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20d { public static double potencia(double base, int exponente) { return Math.pow(base, exponente); } public static double calcularSumaSerie(int numTerminos) { double suma = 0; int base = 2; int exponente = 3; // Comenzamos con el exponente 3 for (int i = 0; i < numTerminos; i++) { double termino = potencia(base, exponente); suma += termino; base++; exponente++; } return suma; } public static void main(String[] args) { Scanner scanner = new Scanner(System.in); System.out.print("Ingrese la cantidad de términos a sumar: "); int numTerminos = scanner.nextInt(); double resultado = calcularSumaSerie(numTerminos); System.out.println("El resultado de la suma de la serie es: " + resultado); scanner.close(); } } </pre>

Corrida

```

run:
Ingrese la cantidad de términos a sumar: 3
El resultado de la suma de la serie es: 1113.0

```

e. $S = x^2 - x^4 + x^6 - x^8 + x^{10} - \dots$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20e { public static double potencia(double base, int exponente) { return Math.pow(base, exponente); } } </pre>

```

    }

    public static double calcularSumaSerie(int numTerminos, double x) {
        double suma = 0;
        int exponente = 2; // Comenzamos con el exponente 2

        for (int i = 0; i < numTerminos; i++) {
            double termino = potencia(x, exponente);
            if (i % 2 == 0) { // Términos pares son positivos
                suma += termino;
            } else { // Términos impares son negativos
                suma -= termino;
            }
            exponente += 2;
        }

        return suma;
    }

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

        System.out.print("Ingrese la cantidad de términos a sumar: ");
        int numTerminos = scanner.nextInt();

        System.out.print("Ingrese el valor de x: ");
        double x = scanner.nextDouble();

        double resultado = calcularSumaSerie(numTerminos, x);

        System.out.println("El resultado de la suma de la serie es: " + resultado);
    }
}

```

Corrida

```

run:
Ingrese la cantidad de términos a sumar: 2
Ingrese el valor de x: 2
El resultado de la suma de la serie es: -12.0

```

f.
$$S = \frac{x}{2} + \frac{x}{4} + \frac{x}{6} + \frac{x}{8} + \frac{x}{10} + \dots$$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20f { public static double calcularSumaSerie(int numTerminos, double x) { double suma = 0; int denominador = 2; // Comenzamos con el denominador 2 for (int i = 0; i < numTerminos; i++) { double termino = x / denominador; suma += termino; denominador += 2; // Incrementamos el denominador en 2 para el siguiente término par } return suma; } public static void main(String[] args) { </pre>

	<pre> Scanner scanner = new Scanner(System.in); System.out.print("Ingrese la cantidad de términos a sumar: "); int numTerminos = scanner.nextInt(); System.out.print("Ingrese el valor de x: "); double x = scanner.nextDouble(); double resultado = calcularSumaSerie(numTerminos, x); System.out.println("El resultado de la suma de la serie es: " + resultado); scanner.close(); } } </pre>
--	--

Corrida:

run:

Ingrese la cantidad de términos a sumar: 2

Ingrese el valor de x: 2

El resultado de la suma de la serie es: 1.5

g. Diseñar un algoritmo que calcule la aproximación del seno:

$$\text{sen}(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \dots$$

Análisis y diseño	Programa
	<pre> package modular; import java.util.Scanner; /** * * @author DELL */ public class Ejer20g { /** * * @author DELL */ // Función para calcular el factorial de un número public static long factorial(int n) { if (n == 0) { return 1; } else { return n * factorial(n - 1); } } // Función para calcular la potencia de un número public static double potencia(double base, int exponente) { return Math.pow(base, exponente); } // Función para calcular la aproximación del seno public static double calcularSeno(double x, int numTerminos) { double seno = 0; for (int i = 0; i < numTerminos; i++) { int exponente = 2 * i + 1; // Exponentes impares int signo = (i % 2 == 0) ? 1 : -1; // Signos alternantes (+, -, +, -) long factorialDenominador = factorial(exponente); double termino = signo * potencia(x, exponente) / factorialDenominador; seno += termino; } return seno; } } </pre>

```
}

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

    System.out.print("Ingrese el valor del ángulo x (en radianes): ");
    double x = scanner.nextDouble();

    System.out.print("Ingrese la cantidad de términos a usar en la aproximación: ");
    int numTerminos = scanner.nextInt();

    double resultado = calcularSeno(x, numTerminos);

    System.out.println("La aproximación del seno de " + x + " es: " + resultado);
    // System.out.println("El valor real de seno(x) es: " + Math.sin(x)); // Para
    comparar

}
```

run:

Ingrese el valor del ángulo x (en radianes): 2

Ingrese la cantidad de términos a usar en la aproximación: 2

La aproximación del seno de 2.0 es: 0.6666666666666667