

## 1. Hallar la suma de los primeros 20 números pares. Desplegar el resultado.

## Análisis

Entrada Proceso				Salida
n	par	suma	i	20
20	0	0	0	
	0	0	1	
	2	2	2	
	4	6	3	
	6	12	4	
	8	20	5	
	10	30	6	
	12	42	7	
	14	56	8	
	16	72	9	
	18	90	10	
	20	110	11	
	22	132	12	
	24	156	13	
	26	182	14	
	28	210	15	
	30	240	16	
	32	272	17	
	34	306	18	
	36	342	19	
	38	380	20	

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; n20[n=20]     n20 --&gt; init[par=0 suma=0]     init --&gt; loop{i=0; i&lt;n; i++}     loop --&gt; process[par=i*2 suma=suma+par]     process --&gt; loop     process --&gt; output[/suma/]     output --&gt; Fin([Fin]) </pre>	<pre> public class Ejer1 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         //         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int n = entrada.nextInt();          int par=0;         int i=0;         par=0;         int suma=0;         for(i=0;i&lt;n;i++)         {             par=i*2;             suma=suma+par;         }         System.out.print( suma );      }  } </pre>

## Corrida:

```

run:
Introduzca N
10
90BUILD SUCCESSFUL

```

## 2. Hallar la suma de los primeros N números impares. desplegar el resultado.

## Análisis

Entrada Proceso				Salida
n	impar	suma	i	25
5	0	0	0	
	1	1	1	
	3	4	2	
	5	9	3	
	7	16	4	
	9	25	5	

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; n[/n/]     n --&gt; Init[impar=0 suma=0]     Init --&gt; Cond{i=0;i&lt;n;i++}     Cond --&gt; Body[impar=i*2+1 suma=suma+impar]     Body --&gt; Cond     Cond --&gt; Suma[/suma/]     Suma --&gt; Fin([Fin]) </pre>	<pre> public class Ejer2 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int n = entrada.nextInt();          int impar=0;         int i=0;         int suma=0;         for(i=0;i&lt;n;i++)         {             impar=i*2+1;             suma=suma+impar;         }         System.out.print( suma );      }  } </pre>

## Corrida:

Consola	
h = 5 25	run: Introduzca N 5 25BUILD SUCCES:

3. Hallar la suma de los pares e impares por separado, comprendidos entre 11 y 101 sin incluir los extremos desplegar los resultados.

## Análisis

Entrada					Proceso		Salida
liminf	limsup	sumapar	sumaimpar	par	impar	i	2520 2564
11	101	0	0	0	0	12	
		12	13			13	
		26	28			14	
		42	45			15	
		60	64			16	
		80	85			17	
		102	108			18	
		126	133			19	
		152	160			20	
		180	189			21	
		210	220			22	
		242	253			23	
		276	288			24	
		312	325			25	
		350	364			26	
		390	405			27	
		432	448			28	
		476	493			29	
		522	540			30	
		570	589			31	
		620	640			32	
		672	693			33	
		726	748			34	
		782	805			35	
		840	864			36	
		900	925			37	
		962	988			38	
		1026	1053			39	
		1092	1120			40	
1092	1120					40	
1160	1189					41	
1230	1260					42	
1302	1333					43	
1376	1408					44	
1452	1485					45	
1530	1564					46	
1610	1645					47	
1692	1728					48	
1776	1813					49	
1862	1900					50	
1950	1989					51	
2040	2080					52	
2132	2173					53	
2226	2268					54	
2322	2365					55	
2420	2464					56	
2520						57	
						91	
						92	
						93	
						94	
						95	
						96	
						97	
						98	
						99	
						100	
						101	

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Init[liminf=11&lt;br/&gt;limsup=101&lt;br/&gt;sumapar=0&lt;br/&gt;sumaimpar=0&lt;br/&gt;par=0&lt;br/&gt;impar=0]     Init --&gt; LoopStart{i=liminf+1; i&lt;limsup; i++}     LoopStart --&gt; Decision{i%2==0}     Decision -- Si --&gt; SumPar[sumapar=sumapar+i]     Decision -- No --&gt; SumImpar[sumaimpar=sumaimpar+i]     SumPar --&gt; Print[/La suma de pares e impares es: "+sumapar+", "+sumaimpar+", respectivamente/]     SumImpar --&gt; Print     Print --&gt; Fin([Fin])   </pre>	<pre> public class Ejer3 {      /**      * @param args the      * command line arguments      */     public static void     main(String[] args) {         // TODO code         application logic here         int liminf=0;         int i=0;         liminf=11;         int limsup=101;         int sumapar=0;         int sumaimpar=0;         int par=0;         int impar=0;         for(i=liminf+1;i&lt;limsup;i++)         {             if(i%2==0)             {                 sumapar=sumapar+i;             }              else             {                 sumaimpar=sumaimpar+i;             }         }         System.out.print( "La suma         de pares e impares es:         "+sumapar+", "+sumaimpar+",         respectivamente" );      }  }   </pre>

**Corrida:****Consola**

La suma de pares e impares es: 2520,2464, respectivamente

run:

La suma de pares e impares es: 2520,2464, respectivamenteB'

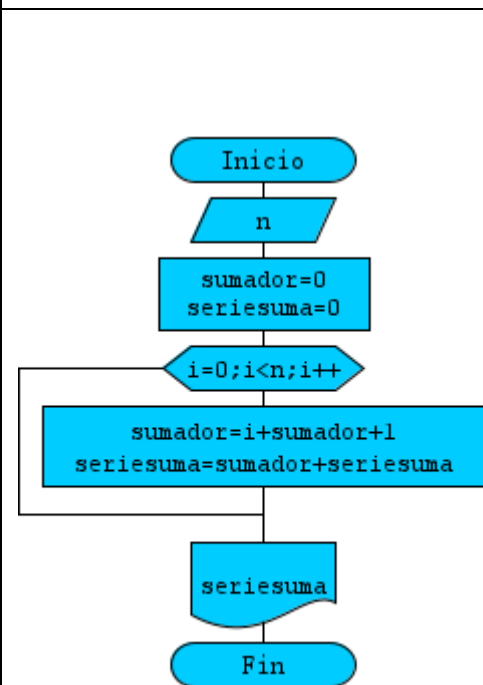
## 4. Evaluar la siguiente sumatoria de N números.

$$S = 1 + 3 + 6 + 10 + 15 + \dots$$

## Análisis

Entrada		Proceso		Salida
n	sumador	seriesuma	i	
3	0	0	0	
	1	1	1	
	3	4	2	
	6	10	3	

## Diseño



## Programa

```

public class Ejer4 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduzca N");
        int n = entrada.nextInt();

        int sumador=0;
        int i=0;

        sumador=0;
        int seriesuma=0;
        for(i=0;i<n;i++)
        {
            sumador=i+sumador+1;
            seriesuma=sumador+seriesuma;
        }
        System.out.print( seriesuma );

    }
}
  
```

## Corrida:

## Consola

```

h = 3
10      run:
        Introduzca N
        3
  
```

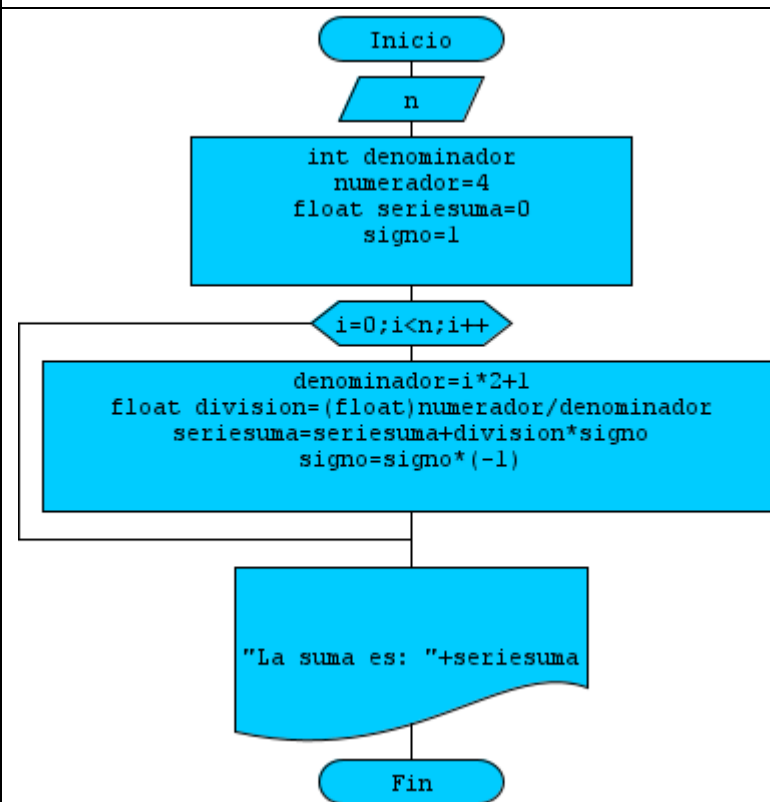
5.

Calcule el valor de  $\pi$ 

$$\pi = \frac{4}{1} - \frac{4}{3} + \frac{4}{5} - \frac{4}{7} + \frac{4}{9} - \dots \dots \dots$$

**Análisis**

Entrada	Proceso	Salida

**Diseño****Programa**

```

Scanner Leer=new Scanner(System.in);
int n;
n = Leer.nextInt();

int denominador;
int numerador=4;

float seriesuma=0;
int signo=1;

int i=0;

for(i=0;i<n;i++)
{
    denominador=i*2+1;
    float division=(float) numerador/denominador;

    seriesuma=seriesuma+division*signo;
    signo=signo*(-1);
}

System.out.print( "La suma es: "+seriesuma );

```

**Corrida:**

```

run:
3
La suma es: 3.4666665

```

6.

Evaluar la siguiente sumatoria de N términos

$$S = \frac{1}{2} + \frac{2}{3} + \frac{3}{4} + \frac{4}{5} + \dots$$

Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre>graph TD     Inicio([Inicio]) --&gt; Input[/n/]     Input --&gt; Init[numerador=1 denominador=2 suma=0 numero=0]     Init --&gt; Cond{i=0; i&lt;n; i++}     Cond --&gt; Process[numero=numerador/denominador; suma=suma+numero numerador++ denominador++]     Process --&gt; Cond     Cond --&gt; Output[/suma/]     Output --&gt; Fin([Fin])</pre>	<pre>Scanner Leer=new Scanner(System.in); int N; N = Leer.nextInt(); int denominador=2; float suma=0; int numerador=1; float numero=0;  for (int i = 0; i &lt; N; i++) {     numero=(float)numerador/denominador;      // System.out.println(numero+"suma");     suma=suma+numero;     numerador=numerador+1;     denominador=denominador+1; }  System.out.println(suma+""); }}</pre>

Corrida:

```
run:
3
1.9166667
```

7. Mostrar la suma de los primeros N números de:  $S = 4 - 8 + 12 - 16 + 20 - \dots$

### Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Input[/n/]     Input --&gt; Cond{n &lt;= 0}     Cond --&gt; Init[serie=4 signo=1 suma=0]     Init --&gt; LoopCond{i=0; i=n; i++}     LoopCond --&gt; Process[suma=suma+serie*signo signo=signo*(-1) serie=serie+4]     Process --&gt; LoopCond     LoopCond --&gt; Output[/suma/]     Output --&gt; Fin([Fin]) </pre>	<pre> Scanner Leer = new Scanner(System.in); int n = 0; int serie = 0; int i = 0; int signo = 1; float suma = 0;  do {     System.out.println("Ingrese un número mayor que 0:");     n = Leer.nextInt(); } while (n &lt;= 0);  serie = 4;  for (i = 0; i &lt; n; i++) {     suma = suma + serie * signo;     signo = signo * (-1);     serie = serie + 4; }  System.out.println("La suma es: " + suma); } } </pre>

### Corrida:

```

run:
Ingrese un número mayor que 0:
3
La suma es: 8.0

```



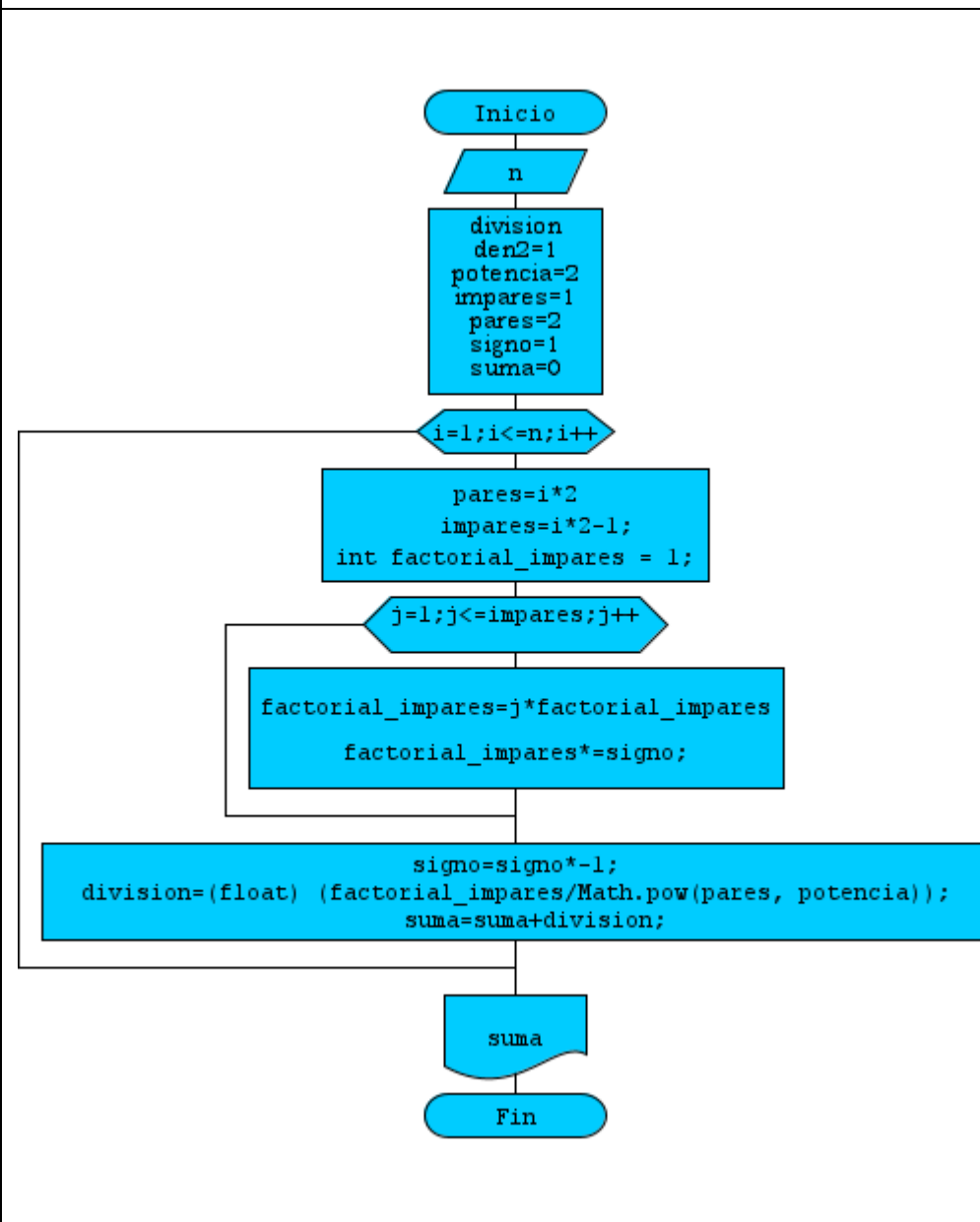
## 8. Evaluar la siguiente sumatoria de N términos

$$S = \frac{1!}{2^2} - \frac{3!}{4^2} + \frac{5!}{6^2} - \frac{7!}{8^2} \pm \dots$$

## Análisis

Entrada	Proceso	Salida

## Diseño



## Programa

```

public static void
main(String[] args) {
    // TODO code application
    logic here
    Scanner entrada = new
    Scanner(System.in);
    int n = entrada.nextInt();
    float suma = 0;
    int signo = 1;
    float division;
    int potencia=2,impares=1,pares=2;
    for (int i =1; i <= n; i++) {
        pares=i*2;
        impares=i*2-1;

        ////////////
        int factorial_impares = 1;
        for (int j = 1; j <= impares;
        j++) {
            factorial_impares *= j;
            factorial_impares*=signo;

        }
        signo=signo*-1;
        // Imprimir el factorial de
        los impares
        // System.out.println("Factorial
        de " + impares + ": " +
        factorial_impares);
        division=(float)
        (factorial_impares/Math.pow(pares,
        potencia));
        suma=suma+division;
    }

    System.out.println("La
    suma es:"+suma);

}
}

```

## Corrida:

```

run:
2
La suma es:-0.125

```

9.

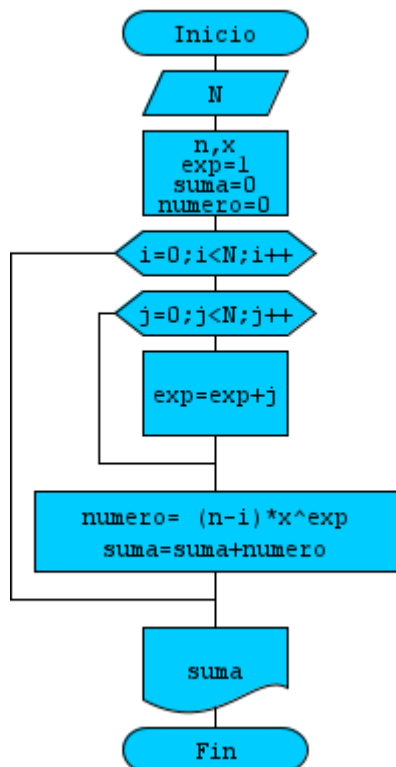
Calcular el valor de S para N términos

$$S = nx + (n-1)x^2 + (n-2)x^3 + (n-3)x^4 + \dots$$

## Análisis

Entrada	Proceso	Salida

## Diseño



## Programa

```

Scanner entrada = new Scanner(System.in);
System.out.println("Introduzca N");
int N = entrada.nextInt();
int n,x,exp=1,suma=0,numero=0;
System.out.println("Introduzca n: ");
n=entrada.nextInt();
System.out.println("Introduzca x: ");
x=entrada.nextInt();
for (int i = 0; i < N; i++) {

    exp=exp+i;
    numero=(int) ((n-i)*Math.pow(x, exp));
    suma=suma+numero;

}System.out.println("La suma es: "+suma); }
  
```

## Corrida:

```

run:
Introduzca N
2
Introduzca n:
3
Introduzca x:
2
La suma es: 14
  
```

10.

Mostrar la suma de los primeros N números de la serie:

$$S = 1 - \frac{1}{2^2} - \frac{1}{3^2} + \frac{1}{4^2} - \frac{1}{5^2} \pm \dots$$

## Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Input[/n/]     Input --&gt; Init[signo=1 serie=1 numero suma=0]     Init --&gt; Loop[i=1; i&lt;=N; i++]     Loop --&gt; Decision{i==serie}     Decision -- No --&gt; SignoNeg[signo=-1]     Decision -- Si --&gt; SignoPos[signo=1 serie=serie+3]     SignoNeg --&gt; Calc[numero=(1/i^2)*signo suma=suma+numero]     SignoPos --&gt; Calc     Calc --&gt; Suma[suma]     Suma --&gt; Fin([Fin]) </pre>	<pre> public class ejer10 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();          int signo = 1;         int serie=1;         float numero;         float suma=0;          for (int i = 1; i &lt;= N; i++) {              if (i==serie) {signo=1; serie=serie+3;}              else{signo=-1;}             //System.out.print(", "+signo);             numero=(float) ((1/Math.pow(i, 2))*signo);             // System.out.print(", "+numero);             suma=suma+numero;          }          System.out.println("La suma es:"+suma);      }  } </pre>

```

run:
Introduzca N
2
La suma es:0.75

```

Corrida:

11.  
Evaluar la siguiente sumatoria de N términos

$$S = \frac{x^0}{2!} - \frac{x^1}{3!} + \frac{x^2}{5!} - \frac{x^3}{7!} \pm \dots$$

Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Input[/n,x/]     Input --&gt; Init[numero, suma=0, contador=0]     Init --&gt; Loop1{i=0; contador&lt;N, i++}     Loop1 --&gt; Signo[signo=1]     Signo --&gt; Div2{i%2!=0}     Div2 -- Si --&gt; SignoNeg[signo=-1]     Div2 -- No --&gt; Div2     SignoNeg --&gt; Div2     Div2 --&gt; DivPrimo[divisor=0, primo=0]     DivPrimo --&gt; Loop2{j=1; j&lt;=i; j++}     Loop2 --&gt; DivJ{i*j==0}     DivJ -- Si --&gt; DivInc[divisor=divisor+1]     DivJ -- No --&gt; DivJ     DivInc --&gt; DivJ     DivJ --&gt; Div2     DivJ --&gt; DivPrimo2[divisor==2]     DivPrimo2 -- Si --&gt; PrimoSet[primo=i&lt;br/&gt;contador++]     DivPrimo2 -- No --&gt; DivPrimo2     PrimoSet --&gt; Factorial[factorial=1]     Factorial --&gt; Loop3{k=1; k&lt;=primo, k++}     Loop3 --&gt; FactMul[factorial*=k]     FactMul --&gt; Loop3     Loop3 --&gt; CalcNumero[numero = (float) (Math.pow(x, i) / factorial)*signo;&lt;br/&gt;suma += numero;]     CalcNumero --&gt; Loop1     CalcNumero --&gt; Suma[/suma/]     Suma --&gt; Fin([Fin])   </pre>	<pre> public class Ejer11 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();         System.out.println("Introduzca x");         int x = entrada.nextInt();         float numero, suma = 0;         ///N números primos         int contador=0;         for (int i = 0; contador &lt; N; i++) {             int signo=1;             if (i%2!=0) {signo=-1;              int divisor=0,primo=0;             for (int j = 1; j &lt;= i; j++) {                 if (i%j==0) {divisor=divisor+1; }                  if (divisor==2) {primo=i;//System.out.print(primo+",")    ;                 contador++;                 // Calcular el factorial del número primo actual                 int factorial = 1;                 for (int k = 1; k &lt;= primo; k++) {                     factorial *= k;                 }                 // Calcular el término y sumarlo                 numero = (float) (Math.pow(x, i) / factorial)*signo;                  System.out.print(numero + ", ");                 suma += numero;              }          }         System.out.println("La suma es: "+suma);      }  }   </pre>

**Corrida:**

```

run:
Introduzca N
2
Introduzca x
1
0.5, -0.16666667, La suma es: 0.3333333
  
```

12.

Hallar la sumatoria de los N primeros términos de:

$$S = 1 \cdot 3 + 3 \cdot 5 + 5 \cdot 7 + 7 \cdot 9 + 9 \cdot 11 + \dots$$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; N[/n/]     N --&gt; Init[aux=3, serie=0 suma=0 impar]     Init --&gt; LoopCond{i=0; i&lt;N; i++}     LoopCond --&gt; LoopBody[impar=2*i+1 serie=aux*impar aux=aux+2 suma=suma+serie]     LoopBody --&gt; LoopCond     LoopCond --&gt; Suma[/suma/]     Suma --&gt; Fin([Fin]) </pre>	<pre> public class ejer12 {     /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();         int aux=3;         int serie=0,suma=0;         for (int i = 0; i &lt; N; i++) {             int impar=2*i+1;             serie=aux*impar;             aux=aux+2;             suma=suma+serie;         }         System.out.println("La suma es: "+suma);     } } </pre>

**Corrida:**

```

run:
Introduzca N
5
La suma es: 215

```

13.

Hallar la suma de los N primeros términos de la siguiente sumatoria

$$S = \frac{1}{1+x} - \frac{1}{1+2x} + \frac{1}{1+3x} - \frac{1}{1+4x} + \frac{1}{1+5x} \pm \dots$$

Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Input[/N, x/]     Input --&gt; Init[signo=1; suma=0; numero]     Init --&gt; Loop[i=1; i&lt;=N; i++]     Loop --&gt; Cond{i%2==0}     Cond -- Si --&gt; SignoNeg[signo=-1]     Cond -- No --&gt; SignoPos[signo=1]     SignoNeg --&gt; CalcNum[numero = (1.0 / (1 + i * x)) * signo;]     SignoPos --&gt; CalcNum     CalcNum --&gt; Suma[suma += numero;]     Suma --&gt; Loop     Suma --&gt; Output[suma]     Output --&gt; Fin([Fin]) </pre>	<pre> public class Ejer13 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();         System.out.println("Introduzca x");         int x = entrada.nextInt();         float signo=1,suma=0;         float numero;         for (int i = 1; i &lt;= N; i++) {             if (i % 2 == 0) {                 signo = -1;             } else {                 signo = 1;             }             numero = (float) (1.0 / (1 + i * x)) * signo; //             División con decimales             suma += numero;         }         System.out.println("La suma es: " + suma);     } } </pre>

Corrida:

```

run:
Introduzca N
2
Introduzca x
2
La suma es: 0.13333334

```

14.

Hallar la suma de los N primeros términos de la siguiente sumatoria

$$S = \frac{1}{2^0} + \frac{1}{2^1} + \frac{1}{2^2} + \frac{1}{2^3} + \frac{1}{2^4} + \dots$$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; N[/N/]     N --&gt; Init[suma=0; numero]     Init --&gt; Loop{i=0; i&lt;N; i++}     Loop --&gt; Calc[numero=(1/(2^i)) suma=suma+numero]     Calc --&gt; Suma[suma]     Suma --&gt; Fin([Fin])     Loop --&gt; Calc </pre>	<pre> public class Ej14 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();         float suma=0;         for (int i = 0; i &lt; N; i++) {             float numero= (float) (1/(Math.pow(2, i)));             suma=suma+numero;         }         System.out.println("La suma es: "+suma);     }  } </pre>

**Corrida:**

```

run:
Introduzca N
2
La suma es: 1.5

```



15.

Hallar la suma de los N primeros términos de la siguiente sumatoria

$$S = \frac{(x+1)!}{x^n} + \frac{(x+2)!}{x^{n-1}} + \frac{(x+3)!}{x^{n-2}} + \frac{(x+4)!}{x^{n-3}} + \frac{(x+5)!}{x^{n-4}} + \frac{(x+6)!}{x^{n-5}} \dots$$

**Análisis**

Entrada	Proceso	Salida
Diseño		Programa
<pre> graph TD     Inicio([Inicio]) --&gt; N[/N/]     N --&gt; Init[x, suma=0, numero, numerador, denominador]     Init --&gt; Loop1{i = 0; i &lt; N; i++}     Loop1 --&gt; CalcDenom[denominador=x^(N-i) numerador=x+i+1 factorial=1]     CalcDenom --&gt; Loop2{k=1; k &lt;= numerador1; k++}     Loop2 --&gt; CalcFact[factorial=factorial*k]     CalcFact --&gt; CalcSum[numero=factorial/denominador; suma=suma+numero]     CalcSum --&gt; Loop1     CalcSum --&gt; Suma[/suma/]     Suma --&gt; Fin([Fin]) </pre>		<pre> public class Ejer15 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         int N = entrada.nextInt();          System.out.println("Introduzca x");         float x = entrada.nextFloat();         float suma=0, numero;         float numerador, denominador;          for (int i = 0; i &lt; N; i++) {             denominador=(float) Math.pow(x, N-i);             int numerador1=(int)x+i+1;//introducir el             factorial              int factorial=1;              //factorial             for (int k = 1; k &lt;= numerador1; k++) {                 factorial=factorial*k;             }             // System.out.println("Factorial de " +             numerador1 + " es: " + factorial);              numero=factorial/denominador;             suma=suma+numero;         }         System.out.println("La suma es: "+suma);      } } </pre>

**Corrida:**

```

run:
Introduzca N
2
Introduzca x
2
La suma es: 13.5

```

16.

Evaluar la siguiente sumatoria para N términos

$$S = \sin T + \sin 3T + \sin 5T + \dots + \sin(2n-1)T = \sum_{i=1}^n \sin(2i-1)T$$

Análisis

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; Input[/N, T/]     Input --&gt; Init[numero, suma=0]     Init --&gt; Loop{i = 1; i &lt;= N; i++}     Loop --&gt; Process[numero=sin((2*i-1)*T) suma=suma+numero]     Process --&gt; Loop     Loop --&gt; Output[suma]     Output --&gt; Fin([Fin]) </pre>	<pre> Scanner entrada = new Scanner(System.in); System.out.println("Introduzca N"); float N = entrada.nextFloat();  System.out.println("Introduzca T"); float T = entrada.nextFloat(); float numero, suma=0; for (int i = 1; i &lt;= N; i++) {     numero = (float) Math.sin(Math.toRadians((2 * i - 1) * T));     suma=suma+numero; } System.out.println("Sin2 es: "+Math.sin(Math.toRadians(2))); System.out.println("Sin6 es: "+Math.sin(Math.toRadians(6))); System.out.println("La suma es: "+suma); </pre>

Corrida:

```

run:
Introduzca N
2
Introduzca T
2
Sin2 es: 0.03489949670250097
Sin6 es: 0.10452846326765347
La suma es: 0.13942796

```

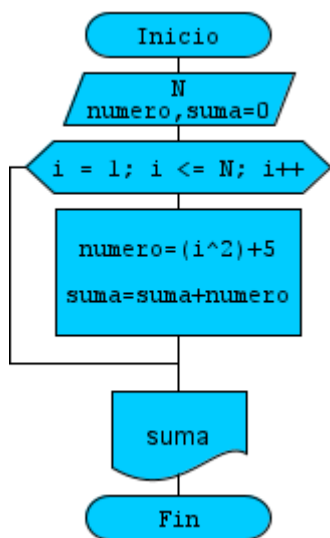
17.

Hallar la suma de los primeros N términos de:

$$\sum_{i=1}^N (i^2 + 5)$$

**Análisis**

Entrada	Proceso	Salida

**Diseño****Programa**

```

public class Ejer17 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduzca N");
        float N = entrada.nextFloat();
        int numero, suma=0;
        for (int i = 1; i <= N; i++) {
            numero=(int) (Math.pow(i, 2)+5);
            suma+=numero;
        }
        System.out.println("La suma es: "+suma);
    }

}
  
```

**Corrida:**

```

run:
Introduzca N
2
La suma es: 15
  
```

18.

Hallar la suma de los N primeros términos de la siguiente sumatoria

$$\sum_{y=1}^N \sum_{x=1}^N \frac{y}{x}$$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; N[/N/]     N --&gt; Init[numero, suma=0]     Init --&gt; Cond{i = 1; i &lt;= N; i++}     Cond --&gt; Process[numero=i/i; suma=suma+numero;]     Process --&gt; Cond     Cond --&gt; Output[/suma/]     Output --&gt; Fin([Fin]) </pre>	<pre> Scanner entrada = new Scanner(System.in); System.out.println("Introduzca N"); float N = entrada.nextFloat();  int numero,suma=0; for (int i = 1; i &lt;= N; i++) {     numero=i/i;     suma=suma+numero; } System.out.println("La suma es:"+suma); } </pre>

**Corrida:**

```

run:
Introduzca N
2
La suma es:2

```

19.

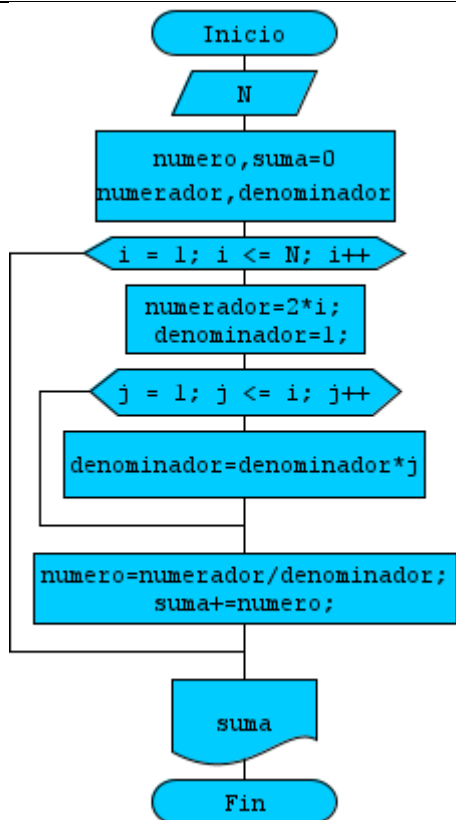
Hallar la suma de los primeros N términos de:

$$\sum_{i=1}^N \frac{2 * i}{i!}$$

## Análisis

Entrada	Proceso	Salida

## Diseño



## Programa

```

public class Ej19 {

    /**
     * @param args the command line arguments
     */
    public static void main(String[] args) {
        // TODO code application logic here
        Scanner entrada = new Scanner(System.in);
        System.out.println("Introduzca N");
        float N = entrada.nextFloat();
        float numero, suma=0, numerador, denominador;
        for (int i = 1; i <= N; i++) {
            numerador=2*i;
            ///factorial
            denominador=1;
            for (int j = 1; j <= i; j++) {
                denominador=denominador*j; }
            //System.out.println("El factorial de: "+i+"es:
            "+denominador);
            numero=numerador/denominador;
            suma+=numero;
        }
        System.out.println("La suma es: "+suma);
    }
}
  
```

## Corrida:

```

run:
Introduzca N
3
La suma es: 5.0
  
```

20.

Hallar la suma de los N primeros términos de la siguiente sumatoria

$$\sum_{y=1}^N \sum_{x=1}^N \frac{y+3}{2*x}$$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
<pre> graph TD     Inicio([Inicio]) --&gt; N[/N/]     N --&gt; Init[numerador, denominador numero,suma=0]     Init --&gt; Loop{i = 1; i &lt;= N; i++}     Loop --&gt; Process[numerador=i+3; denominador=2*i; numero=numerador/denominador; suma=suma+numero;]     Process --&gt; Loop     Process --&gt; Suma[suma]     Suma --&gt; Fin([Fin]) </pre>	<pre> public class Ejer20 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);         System.out.println("Introduzca N");         float N = entrada.nextFloat();         float numerador,denominador,numero,suma=0;         for (float i = 1; i &lt;= N; i++) {             numerador=i+3;             denominador=2*i;             numero=numerador/denominador;             suma=suma+numero;         }         System.out.println("La suma es:"+suma);     } } </pre>

**Corrida:**

```

run:
Introduzca N
3
La suma es: 5.0

```

1. Dado un número natural E, formar otro número con los dos últimos dígitos.  
Ejemplo Si E=3421 el número resultante es 21

**Análisis**

	Salida
<b>Entrada Proceso</b>	

Diseño	Programa
	<pre> public class Ejer1 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         int n,dig;         Scanner entrada=new Scanner(System.in);         do {             System.out.println("Ingresar número");             n=entrada.nextInt();         } while (n&lt;=0);          dig= n % 100; // Obtenemos los dos últimos digitos          System.out.println("Los dos últimos dígitos son: " + dig);     }  } </pre>

**Corrida:**

```

run:
Ingresar número
5654
Los dos últimos digitos son: 54

```

2. Dado un número X entero positivo reemplazar los dígitos pares superiores (en el caso del 9 reemplazar por 0).

Ejemplo Si X=6529 Entonces 8640

### Análisis

	Salida
Entrada Proceso	

Diseño	Programa
	<pre> public class Ejer2 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // Dado un número X entero positivo reemplazar los dígitos pares superiores (en el caso         // del 9         // reemplazar por 0).         // Ejemplo Si X=6529 □ 8640         int n,dig,nn=0,p=1;         Scanner entrada=new Scanner(System.in);         do {             System.out.println("Ingresar número:");             n = entrada.nextInt();         } while (n &lt;= 0);          while (n &gt; 0) {             dig = n % 10;             n = n / 10;              if (dig % 2 == 0) {                 dig += 2;             } else {                 dig++;             }              if (dig == 10) { // Si el dígito es 8 + 2 o 9 + 1, establecer a 0                 dig = 0;             }              nn = nn + dig * p;             p *= 10;         }          System.out.println("El nuevo número es: " + nn);     } </pre>

### Corrida:

```

run:
Ingresar número:
6529
El nuevo número es: 8640

```



## 3. Eliminar los dígitos que se encuentran en las posiciones impares de un número.

## Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer3 {     public static void main(String[] args) {          int n, resultado = 0, lugar = 1;         Scanner entrada = new Scanner(System.in);          do {             System.out.println("Ingresar número:");             n = entrada.nextInt();         } while (n &lt;= 0);          int posición = 1;          while (n &gt; 0) {             int digit = n % 10;             n = n / 10;              if (posición % 2 == 0) {                 resultado += digit * lugar;                 lugar *= 10;             }              posición++; // Incrementamos la posición         }          System.out.println("El nuevo número es: " +             resultado);     } } </pre>

## Corrida:

```

run:
Ingresar número:
564645
El nuevo número es: 544

```

4. Leer un número X y eliminar todos sus dígitos que se encuentren en otro número E previamente leído.

Ejemplo Si X=542389 el número leído E=356 el número resultante es 4289

#### Análisis

Entrada	Proceso	Salida
Diseño	Programa	
	<pre> public class Ejer4 {     public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);         System.out.println("Ingresar número E:");         int E = entrada.nextInt();         int X;          do {             System.out.println("Ingresar número:");             X = entrada.nextInt();         } while (X &lt;= 0);          int resultado = 0;         int lugar = 1;          while (X &gt; 0) {             int digX = X % 10;             X /= 10;              int tempE = E;             boolean estaEnE = false;             while (tempE &gt; 0) {                 int digE = tempE % 10;                 if (digE == digX) {                     estaEnE = true;                     break;                 }                 tempE /= 10;             }             if (!estaEnE) {                 resultado += digX * lugar;                 lugar *= 10;             }         }          System.out.println("El nuevo número es: " +             resultado);     } } </pre>	

```

run:
Ingresar número E:
356
Ingresar número:
542389
El nuevo número es: 4289

```

**Corrida:**

5. Se dice que un número es CUBO INTERESANTE si puede ser representado como la suma de los cubos de sus dígitos. Mostrar en pantalla los 4 primeros cubos interesantes entre todos los enteros mayores a 1. Uno de esos números es 153 pues:  $153 = 1^3 + 5^3 + 3^3 = 1 + 125 + 27 = 153$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre>int contador = 0; int numero = 2; // Empezamos desde el número 2 (mayor a 1)  while (contador &lt; 4) {     int original = numero;     int sumaCubos = 0;     int temp = numero;      while (temp &gt; 0) {         int digito = temp % 10;         sumaCubos += digito * digito * digito;         temp /= 10;     }      if (sumaCubos == original) {         System.out.println(numero);         contador++;     }      numero++; }</pre>

**Corrida:**

```
run:
153
370
371
407
```

6.

Lea un entero n y un dígito d se debe invertir el número n y si d aparece en n se debe reemplazar por el valor 1 en la posición en la que se encuentra d. Por ejemplo: si n=23668 y d=6. Se desplegaría 81132. Además se debe imprimir: Cuantos números se reemplazaron y cuantos no se reemplazaron. El aplicativo también debe informar al usuario de cuantas cifras era el número entero n leído

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer6 {      public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);         int n;         do {             System.out.println("Ingresar número:");             n = entrada.nextInt();         } while (n &lt;= 0);         System.out.println("Ingrese el dígito d:");         int d = entrada.nextInt();          int nInvertido = 0;         int resultado = 0;         int cifrasOriginal = 0;         int cifrasActuales = 0;         int reemplazados = 0;         int noReemplazados = 0;          while (n &gt; 0) {             nInvertido = nInvertido * 10 + n % 10;             n /= 10;             cifrasOriginal++;         }          int posicion = 1;         while (nInvertido &gt; 0) {             int digito = nInvertido % 10;             nInvertido /= 10;             cifrasActuales++;              if (digito == d) {                 resultado += 1 * posicion;                 reemplazados++;             }         }     } } </pre>

```
        } else {
            resultado += digito * posicion;
            noReemplazados++;
        }
        posicion *= 10;
    }
    System.out.println("Número invertido y con
reemplazo: " + resultado);
    System.out.println("Cifras del número
original n: " + cifrasOriginal);
    System.out.println("Números reemplazados: "
+ reemplazados);
    System.out.println("Números no
reemplazados: " + noReemplazados);
    }
}
```

**Corrida:**

run:

Ingrese el número entero n:

23668

Ingrese el dígito d:

6

Número invertido y con reemplazo: 81132

Cifras del número original n: 5

Números reemplazados: 2

Números no reemplazados: 3

7. Pedir un número entero positivo mayor a 100, intercambiar el primer dígito con el último y mostrar el nuevo número. Ejemplo si el número es 457 se muestra 754.

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer7 {     Scanner entrada = new Scanner(System.in);     int numero;     do {         System.out.println("Ingrese un número entero positivo mayor a 100:");         numero = entrada.nextInt();     } while (numero &lt;= 100);      int primerDigito = numero;     int ultimoDigito = numero % 10;     int cifras = 1;      while (primerDigito &gt;= 10) {         primerDigito /= 10;         cifras *= 10;     }      int nuevoNumero = ultimoDigito * cifras + (numero % cifras / 10) * 10 + primerDigito;      System.out.println("El nuevo número es: " + nuevoNumero);     } } </pre>

**Corrida:**

```

run:
Ingrese un número entero positivo mayor a 100:
457
El nuevo número es: 754

```

8. Dado un numero X, hacer rotar hacia la derecha en K dígitos, desplegar el numero obtenido.

## Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer8 {      /**      * @param args the command line arguments      */     public static void main(String[] args) {         // Dado un numero X, hacer rotar hacia la derecha en K         dígitos, desplegar el numero obtenido.         Scanner entrada = new Scanner(System.in);         int x,k,de,d,xx,p;         do {             System.out.println("Ingresar número:");             x = entrada.nextInt();         } while (x &lt;= 0);         System.out.println("Ingrese k");   k=entrada.nextInt();          for (int i = 1; i &lt;= k; i++) {             de=x%10;             x=x/10;             xx=0;             p=1;             while (x!=0) {                 d=x%10;                 x=x/10;                 xx=d*p+xx;                 p=p*10;             }             xx=de*p+xx;             x=xx;         }         System.out.println("El número obtenido es: "+x);     }  } </pre>

## Corrida:

```

run:
Ingresar número:
9543
Ingrese k
2
El número obtenido es: 4395

```

9.

Calcular el valor de S para N términos

$$S = nx + (n-1)x^2 + (n-2)x^3 + (n-3)x^4 + \dots$$

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer9 {      public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);         System.out.println("Ingrese un número Z mayor a 0:");         int Z = entrada.nextInt();          System.out.println("Ingrese un número A mayor a 0:");         int A = entrada.nextInt();          int nuevoNumero = 0;         int posicion = 1;          while (Z &gt; 0) {             int digito = Z % 10;             Z /= 10;             if (digito != A) {                 nuevoNumero = digito * posicion + nuevoNumero;                 posicion *= 10;             }         }          System.out.println("El nuevo número Z es: " + nuevoNumero);     } } </pre>

**Corrida:**

```

run:
Ingrese un número Z mayor a 0:
85687884
Ingrese un número A mayor a 0:
8
El nuevo número Z es: 5674

```



10.

Dado un número natural  $n$  encuentre su raíz digital. Raíz digital de un natural: se calcula el natural  $m$  sumando los dígitos que componen a  $n$ . El proceso se repite sobre el nuevo número hasta que el resultado sea de un dígito.

Ejemplo:  $347 \rightarrow 3 + 4 + 7 = 14 \rightarrow 1 + 4 = 5 \rightarrow RD(347) = 5$

Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer10 {     public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);         System.out.println("Ingrese un número natural n:");         int n = entrada.nextInt();          while (n &gt;= 10) {             int sumaDigitos = 0;             while (n &gt; 0) {                 sumaDigitos += n % 10;                 n /= 10;             }             n = sumaDigitos;         }          System.out.println("La raíz digital es: " + n);     } } </pre>

Corrida:

run:

Ingrese un número natural n:

347

La raíz digital es: 5

11.

Dado un número N entero positivo, añadir los dígitos que sean necesarios a N para obtener un nuevo número capicúa en M.

Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer11 {      public static void main(String[] args) {          Scanner entrada = new Scanner(System.in);          System.out.println("Ingrese un número entero positivo N:");         int N = entrada.nextInt();         int original = N;         int reverso = 0;          while (N &gt; 0) {             int digito = N % 10;             reverso = reverso * 10 + digito;             N /= 10;         }          int longitud = (int) Math.log10(original) + 1;          int capicua = original * (int) Math.pow(10, longitud) + reverso;          System.out.println("El número capicúa M es: " + capicua);     } } </pre>

Corrida:

run:

Ingrese un número entero positivo N:

4568

El número capicúa M es: 45688654

BUILD SUCCESSFUL (total time: 7 seconds)

12.

Dado un número Z entero y positivo, eliminar aquellos dígitos que se encuentren en las posiciones impares. Mostrar el número resultante.  
Ejemplo: para Z= 48253 entonces NZ= 85

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer12 {     public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);          System.out.println("Ingrese un número entero positivo Z:");         int Z = entrada.nextInt();          int NZ = 0, p=1;         int posicion = 1; // Inicializar la posición del dígito en Z          while (Z &gt; 0) {             int digito = Z % 10;             Z /= 10;              if (posicion % 2 == 0) {                 NZ = NZ + digito*p;                 p=p*10;             }              posicion++;         }          System.out.println("El número resultante NZ es: " + NZ);     } } </pre>

**Corrida:**

```

run:
Ingrese un número entero positivo Z:
48253
El número resultante NZ es: 85

```

13. 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 de A y B, que también deberá estar ordenado ascendentemente.

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa

**Corrida:**

14.

Dado un numero n positivo entero, determinar la cantidad de dígitos cero que contenga.  
Ejemplo Si n=102340 en número contiene dos dígitos cero

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> */ public class Ejer14 {      public static void main(String[] args) {         Scanner entrada = new Scanner(System.in);          System.out.println("Ingrese un número entero positivo:");         int n = entrada.nextInt();          int contadorCeros = 0;          while (n &gt; 0) {             // Obtener el último dígito             int digito = n % 10;              if (digito == 0) {                 contadorCeros++;             }              n /= 10;         }          System.out.println("El número contiene " + contadorCeros             + " dígitos cero.");     } } </pre>

**Corrida:**

```

run:
Ingrese un número entero positivo:
123000
El número contiene 3 dígitos cero.

```

15.

Leer un número Z entero y positivo. Eliminar aquellos dígitos cuyo valor sea igual al menor de ellos.

Ejemplo: Z= 3453733 Dígito menor es=3 Sale: NZ=457

**Análisis**

Entrada	Proceso	Salida
Diseño		Programa
		<pre> Scanner entrada = new Scanner(System.in);          System.out.println("Ingrese un número entero positivo Z:");         int Z = entrada.nextInt();          int digitoMenor = 9; // Inicializar con el mayor dígito posible (9)          int numeroTemporal = Z;         while (numeroTemporal &gt; 0) {             int digito = numeroTemporal % 10;             if (digito &lt; digitoMenor) {                 digitoMenor = digito;             }             numeroTemporal /= 10;         }          int NZ = 0;         int potencia = 1;          while (Z &gt; 0) {             int digito = Z % 10;             Z /= 10;              if (digito != digitoMenor) {                 NZ += digito * potencia;                 potencia *= 10;             }         }          System.out.println("El número resultante NZ es: " + NZ);      } </pre>

run:

Ingrese un número entero positivo Z:

3453733

El número resultante NZ es: 457

**Corrida:**

16. Dado un número entero positivo  $n$  se desea saber:

- cuantos de sus dígitos son impares
- cuantos de sus dígitos son múltiplos de dos
- cuantos son cero
- la suma de todos sus dígitos pares

### Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> Scanner entrada = new Scanner(System.in);  System.out.println("Ingrese un número entero positivo n:"); int n = entrada.nextInt();  int impares = 0; int multiplosDeDos = 0; int ceros = 0; int sumaPares = 0;  while (n &gt; 0) {     int digito = n % 10; // Obtener el último dígito     n /= 10; // Eliminar el último dígito      if (digito % 2 != 0) {         impares++;     }      if (digito != 0 &amp;&amp; digito % 2 == 0) {         multiplosDeDos++;         sumaPares += digito; // Sumar dígitos pares     }      if (digito == 0) {         ceros++;     } }  System.out.println("a. Dígitos impares: " + impares); System.out.println("b. Dígitos múltiplos de dos: " + multiplosDeDos); System.out.println("c. Dígitos cero: " + ceros); System.out.println("d. Suma de dígitos pares: " + sumaPares);  } </pre>

### Corrida:

```

run:
Ingrese un número entero positivo n:
564
a. Dígitos impares: 1
b. Dígitos múltiplos de dos: 2
c. Dígitos cero: 0
d. Suma de dígitos pares: 10

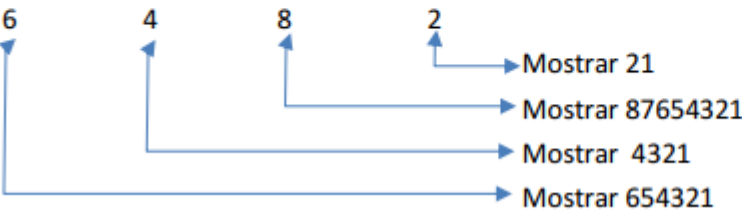
```

17. 17. Dado un número N entero y positivo, mostrar los dígitos pares de N y generar nuevos números con los dígitos decrecientes.

Ejemplo:

N=6458732

Dígitos pares



Análisis

Entrada	Proceso	Salida

Diseño	Programa

Corrida:



18.

## Análisis

Entrada	Proceso	Salida

Diseño	Programa
	<pre> public class Ejer18 {      public static void main(String[] args) {         // TODO code application logic here         Scanner entrada = new Scanner(System.in);          System.out.println("Ingrese un número entero positivo X:");         int X = entrada.nextInt();         entrada.close(); // Cerrar el scanner después de leer la entrada          System.out.println("X=" + X);         while (X &gt; 0) {             // Reducir cada dígito de X en 1 (si es mayor a 0) int nuevoX = 0;             int potencia = 1;             boolean todosCeros = true;              while (X &gt; 0) {                 int digito = X % 10;                 if (digito &gt; 0) {                     digito--; // Disminuir el dígito en 1                     todosCeros = false;                 }                 nuevoX = nuevoX + digito * potencia;                 X /= 10;                 potencia *= 10;             }              X = nuevoX;              if (!todosCeros) {                 System.out.println("X=" + X);             } else {                 break;             }         }     } } </pre>

```

run:
Ingrese un número entero positivo X:
3687
X=3687
X=2576
X=1465
X=354
X=243
X=132
X=21
X=10
X=0

```

Corrida: -----

19.

Dado un número real positivo  $r$  se pide invertir los dígitos del numero manteniendo la cantidad de dígitos en la parte entera como en la decimal  
Ejemplo Si  $r=123.45$   $s=543.21$

Análisis

Entrada	Proceso	Salida

Diseño	Programa

Corrida:

20.

Dado un número entero positivo A eliminar todos los dígitos k el valor de k esta entre 0 y 9.

Ejemplo A=48566462 y k= 6 Sale: NA= 48542

**Análisis**

Entrada	Proceso	Salida

Diseño	Programa
	<pre> Scanner entrada = new Scanner(System.in); System.out.println("Ingrese un número entero positivo A:"); int A = entrada.nextInt(); System.out.println("Ingrese el dígito k a eliminar (entre 0 y 9):"); int k = entrada.nextInt(); int NA = 0; int potencia = 1; // Potencia de 10 para extraer cada dígito de A while (A &gt; 0) {     int digito = A % 10; // Obtener el último dígito de A     A /= 10; // Eliminar el último dígito de A      if (digito != k) {         NA = digito * potencia + NA;         potencia *= 10; // Incrementar la potencia de 10 para el próximo dígito     } }  System.out.println("El número resultante NA es: " + NA); } </pre>

**Corrida:**

```
run:
```

```
Ingrese un número entero positivo A:
```

```
48566462
```

```
Ingrese el dígito k a eliminar (entre 0 y 9):
```

```
6
```

```
El número resultante NA es: 48542
```