

Series Numéricas

S1: $\begin{matrix} a & b & c & & & & \\ 0 & 1 & 1 & 2 & 3 & 5 & 8 & 13 & \dots \end{matrix}$

La serie sigue la secuencia de la suma de dos números da el siguiente.

$$0+1=1$$

$$1+1=2$$

$$2+1=3 \dots$$

$$\therefore a+b=c$$

$$a=b$$

$$b=c$$

S6: $\begin{matrix} a & & & & & & \\ 1 & 4 & 9 & 16 & 25 & 36 & 49 & 64 & \dots \end{matrix}$

$+3 \quad +5 \quad +7 \quad +9 \quad +11 \quad +13 \quad +15$

La serie aumento con números impares

```
# impar -> (n x 2) + 1 | a = 1
                        | for (int i = 0; i < n; i++)
                        | a = a + (i x 2) + 3
                        |
```

* Series de Caracteres.

S1: $\begin{matrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 \\ + & - & + & - & + & - & + & - & + \dots \end{matrix}$

$(-)$ -> # impares $(+)$ -> # pares

```
\therefore for (int i = 0; i < n; i++)
    if (i % 2 == 0)
        System.out.print (" + ")
    else
        System.out.print (" - ")
```

$\frac{3}{1} \frac{12}{1} \rightarrow \text{impar}$

$\frac{4}{0} \frac{12}{2} \rightarrow \text{par}$

56-1

97 98 99 ..
a b c d e f g h..

Secuencia del abecedario.

* ASCII

```
for (int i = 0; i < h; i++)
```

x = "a" + (i) ; → Cambio de int a char

```
System.out.print ((char) (x) + " ")
```

Sin embargo a ... z → 26 veces. ∴ la serie debe repetirse las veces que se necesario.

∴ $x = "a" + (i \% 26)$

(5) $\begin{array}{r} 26 \\ 0 \end{array}$ f $\begin{array}{r} 31 \\ 5 \end{array}$ $\begin{array}{r} 26 \\ 1 \end{array}$

56-2

a + c - e + g - ...
0 1 2 3 4 5 6 7

pares (letras)

impares (+) || (-)

(+) 1 - 5 -
(-) 3 - 7 -

$\begin{array}{r} -1 \\ 4 \\ 0 \end{array}$

$\begin{array}{r} 5 \\ 1 \\ 1 \end{array}$

```
for (int i = 0; i < h; i++)
```

```
if (i % 2 == 0) → pares
```

```
System.out.print ((char) ('a' + (i % 26) + " ")
```

```
} else if (i % 4 == 1)
```

```
System.out.print (" + ")
```

```
else
```

```
System.out.print (" - ")
```


Cadena de Caracteres

C03:

Eliminar los vocales de una frase

Usamos el método del String.

String sinVocales = frase.replaceAll("a,e,i,o,u,A,E,I,O,U","")

este método reemplaza la condición (letras) que ponemos por la segunda parte de los ""

C08:

Anagrama de palabras.

- Utilizaremos MAP para almacenar la palabra y el anagrama
- HashMap para almacenar la anterior

palabraAnagrama.put("deliro", "lidero");

↳ agrega una clave y un valor

- Pediremos que ingrese un anagrama del listado de palabras
- Además también hacemos uso de:

.get(key) → Obtener el valor asociado a una clave
.containsKey(key) → Verifica si la clave existe en Map.

Arrays

Matriz con el nombre y apellido

Nombre → Monserrath
Apellido → Rodríguez] almacenar su longitud

- Crear una matriz chor

chor [][] matriz = new chor [l.nombre][l.apellido]

```
for (int f = 0; f < l.nombre.length(); f++)  
    for (int c = 0; c < l.apellido.length(); c++)  
        if (f == c)  
            matriz[f][c] = nombre.chor(f)  
        if (f + c == l.apellido.length() - 1)  
            matriz[f][c] = apellido.chor(f)
```



```
else  
    matriz[f][c] = " ";
```

Para imprimir usamos el mismo metodo cambiando (matriz) por Sys.out.

Landing

L04 Waiting 0 a 100% → 0.00 100%.

- Con un for llegaremos al 100.

```
for (int i = 0; i <= 100; i += 10)
```

- Limpiar la consola. `lr`

- Para que aparezca 0.00 y despues 0.00 usaremos %.
para cuando sea por o impar

impar → 0.00
y utilizamos un if

por → 0.00

```
if ( (i/10) % 2 == 0 )  
    Sys.out.print ("0.00 + " + i + "%")
```

else

```
    Sys.out.print ("0.00 + " + i + "%")
```

- Para que se demore un tiempo para que se imprima en consola ocupamos el metodo

```
Thread.sleep ( )
```

L09 Pedir y mostrar nombre completo.

- Debemos conocer la longitud del nombre.

```
for (int i = 0; i < n; i++)
```

porcentaje = (i * 100) / (n - 1) // el indice siempre es desde 0 por lo que debemos reanfor.

```
for (int j = 0; j < i; j++)
```

```
    Sys.out.print (" ")
```

```
}
```

```
Sys.out.print (nombreCompleto.charAt(i) + porcentaje)
```


Figuras

F2:

	1	2	3	4	5
1	*	+	*	+	*
2	+				+
3	*				*
4	+				+
5	*	+	*	+	*

tamaño = 5

C = columna f = fila

$(f == 1 \wedge c == 1) (f == 1 \wedge c == 5)$
 $(f == 3 \wedge c == 1) (f == 3 \wedge c == 5)$
 $(f == 5 \wedge c == 1) (f == 5 \wedge c == 5)$
 $(f == 1 \wedge c == 3) (f == 5 \wedge c == 3)$

```
for (int c=0; c<t; c++) {
```

```
  for (int f=0; f<t; f++) {
```

```
    IF (condiciones (*))
```

```
      Sys.out.print ("*")
```

```
    else IF (condiciones (+))
```

```
      Sys.out.print ("+")
```

```
  else
```

```
    Sys.out.print (" ")
```

```
  }
```

```
  Sys.out.println (" ")
```

```
}
```

$(f == 1 \wedge c == 2) (f == 1 \wedge c == 4)$
 $(f == 2 \wedge c == 1) (f == 2 \wedge c == 5)$
 $(f == 4 \wedge c == 1) (f == 4 \wedge c == 5)$
 $(f == 5 \wedge c == 2) (f == 5 \wedge c == 4)$

tamaño = 5

↳ +

F7:

	1	2	3
1			
2			
3			

tamaño = desplazamiento = d

columnas = c

filas = f

fillos == columnas

```
for (int f=0; f<d; f++)
```

```
  for (int c=0; c<d; c++)
```

```
    if (fillos == columnas)
```

```
      Sys.out.print ("|_")
```

```
    else
```

```
      Sys.out.print (" ")
```

```
  }
```

```
  Sys.out.println (" ")
```

```
}
```


F12

1 2 3 4 5
 1 2 3 4
 1 2 3
 1 2
 1

* Se reduce

(columnas $\leq n - filas + 1$)

* ejemplo

(columnas $\leq 5 - 1 + 1$)

columnas ≤ 5

imprime el # de columnas 1-5

```
for (int f = 1; f <= n; f++)
  for (int c = 1; c <= n; c++)
    if (columnas <= n - filas + 1)
      Sys.out.print (columnas)
    else
      Sys.out.print (" ")
  }
  Sys.out.print (n (" "))
}
```

F17

1 2 3 4 5
 5 1 0 0 1
 4 0 1 0 0
 3 1 0 0 1
 2 1 0 0 1
 1 1 0 0 1

tamaño = 5

Linea p = (filas == columnas)

Linea s = (filas + columna == n + 1)

```
for (int f = 1; filas <= n; filas++)
  for (int c = 1; co <= n; c++)
    if (f == c || f + c == n + 1)
      if (f % 2 == 1)
        Sys.out.print ("1")
      else
        Sys.out.print ("0")
    else
      Sys.out.print (" ")
  }
  Sys.out.print (n (" "))
}
```

	c				
n	1	2	3	4	5
f	5	1			x
4				x	
3			x		
2		x			
1	x				

(f % 2 == 1)

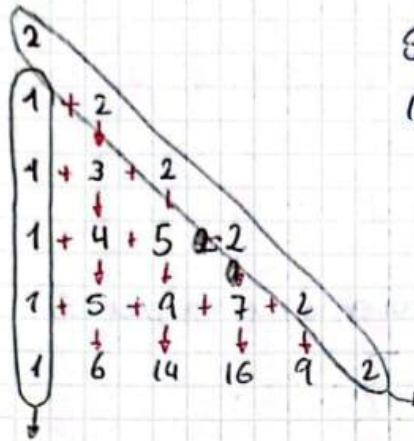
3
1

2

2
0

2
1

F18



Sigue la lógica del triángulo de pascal
la suma de los dos # superiores da el de abajo

Columna == 1

for (int f=0; f<longitud; f++) ← filas

for (int c=0; c<longitud; c++) ← columnas.

if (c == 1) {

System.out.print("1")

} else if (c == f) {

System.out.print("2")

} else {

(f-1, c-1) + (f-1, c)

puedo ponerle en un método
para llamarlo

F3

	0	1	2	3
0 *	*			
1 *	*	*		
2 *	*	*	*	
3 *	*	*	*	*

columnas siempre serán \leq Filas.

for (int f=0; f<= tamaño; f++)

for (int c=0; c<= tamaño; c++)

if (columnas <= filas) {

System.out.print("*")

} else {

System.out.print(" ")

Arroy 5.

Matriz nombre completo random.

1) Pedir nombre completo

- Quitar los espacios
- Crear una matriz con la longitud

2) Obtener posición aleatoria

3) Tener en cuenta que 2 letras pueden estar en la misma posición

- if (matriz [a] [b] != 0)

matriz [a] [b] = "x";

} else {

matriz [a] [b] = letraArray [i]

4) Dejar espacios vacíos donde no hay letras

if (matriz [a] [b] == 0)

matriz [a] [b] = " "

5) Imprimir matriz

```
for (char c : Pila:matriz) {
```

```
for (char c : Pila) {
```

```
System.out.print (c + " ")
```

```
}
```

```
System.out.println ();
```


Recursion

R03 multiplicación (a,b)

1) Ingreso de dos valores a,b.

```
if (b == 0) {
```

```
    Sys.out.print ("0")
```

```
} if (b > 0) {
```

```
    Sys.out.print (a + metodo (a, b-1))
```

```
else {
```

```
    Sys.out.print (-metodo (a, -b));
```


Recursión

R02: suma de (a, b)

• de un arreglo debemos hacer que se sumen los datos

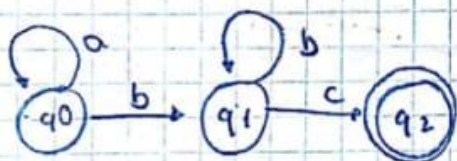
$A = [0, 5, 7, 8, 10]$

Tener en cuenta salir del arreglo cuando ya se haya recorrido todo a traves de

suma (A, i) $\left[\begin{array}{l} \text{if } (i == A.length) \\ \text{return } 0. \\ \text{else} \\ \text{return } A[i] + \text{suma}(A, i+1) \end{array} \right.$

Grafos y automatos.

A01: a^*b+c



	a	b	c
0	0	1	e
1	e	1	2
2	e	e	e

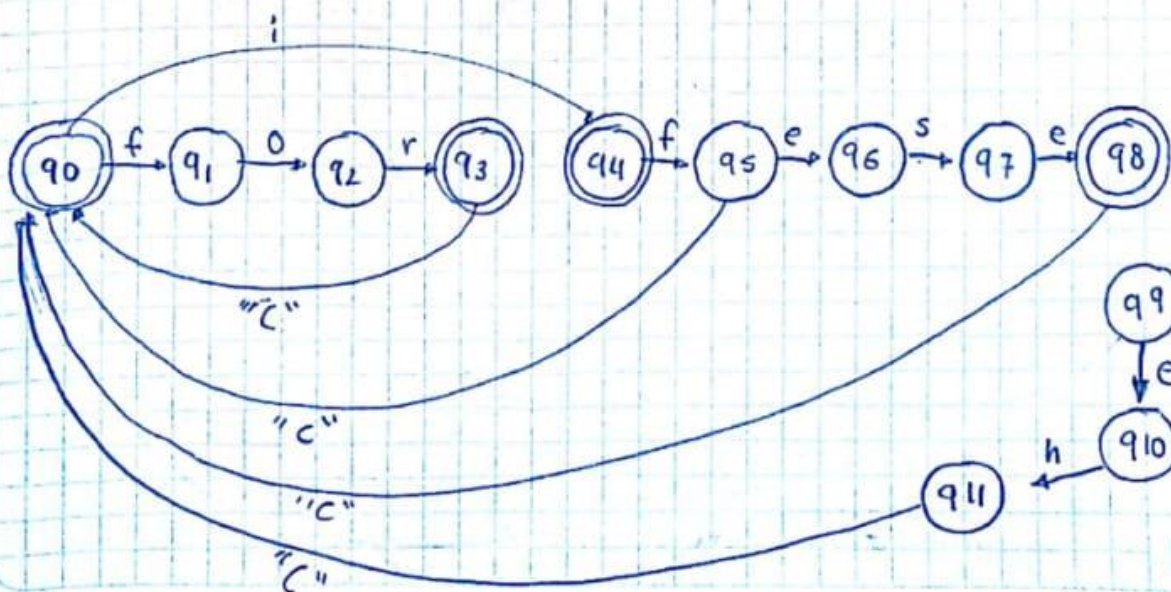
a) se puede poner la "a" que sea o simplemente no

b) "b" si quiero debe estar una vez

c) "c" debe terminar con una

$\Sigma = \{a, b, c\}$
 $Q = \{q_0, q_1, q_2\}$
 $F = \{q_2\}$
 $\delta = \{(q_0, a) q_0; (q_0, b) q_1\}$
 $\{(q_1, b) q_1; (q_1, c) q_2\}$
 $L = \{a^*b+c\}$

A06 = $L = \{for, if, else, ifelse, foreach\}$



$\Sigma = \{ i, p, else, i, p, each, f, p, r \}$

$Q = \{ q_0, q_1, q_2, q_3, q_4, q_5, q_6, q_7, q_8, q_9, q_{10}, q_{11}, q_{12} \}$

$F = \{ q_8, q_4, q_3, q_{11} \}$

$\delta' = \{ q_0, q_1, q_2, \dots \}$

$L = \{ f, o, r, else, i, p, each, else, i, p, else \}$

	f	o	r	i	c	e	s	h
0	1	-	-	4	-	-	-	-
1	-	2	-	-	-	-	-	-
2	-	-	3	-	-	-	-	-
3	-	-	-	-	0	-	-	-
4	5	-	-	-	-	-	-	-
5	-	-	-	-	0	6	-	-
6	-	-	-	-	-	-	7	-
7	-	-	-	-	-	8	-	-
8	-	-	-	-	0	-	-	-
9	-	-	-	-	-	10	-	-
10	-	-	-	-	-	-	-	11
11	-	-	-	-	0	-	-	-