

Trabajo Grupo 3

→ Anahi Pillajo

secuencias

$$S_3: \frac{0}{1}, \frac{1}{3}, \frac{1}{5}, \frac{2}{7}, \frac{3}{9}, \frac{5}{11}, \frac{8}{13}, \frac{13}{15}, \dots ?$$

número dñs: secuencia de Fibonacci cada número es la suma de los dos anteriores

dénominador: números impares de dos en 2
 $(2n-1) \rightarrow$

$$? = \frac{21}{17} //$$

$$\frac{F_n}{(2n-1)}$$

$$S_4: 3, 8, 13, 18, 23, 28, 33, 38, ?$$

? 42 → cada número anterior se le suma 5 //

Series de caracteres

$$S_5: \begin{matrix} 2 & 3 & 5 & 7 & 11 & 13 \\ ++ & +++ & +++++ & ++++++ & +++++++ & +++++++ \\ \backslash & \backslash & \backslash & \backslash & \backslash & \backslash \\ 1 & 2 & 2 & 4 & 2 & \rightarrow ? \text{ no siguen} \\ & & & & & \text{patrón} \end{matrix}$$

→ representa los números primos →

$$? = + + + + + + + + + + + + + + + .$$

$$S_6: a bbb ccccc dddddd eeeeeeee ?$$

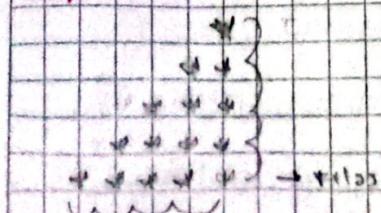
sigue creciendo con números impares

$$(2n-1) \quad ? = f f f f f f f f f f f f //$$

Figuras

$n = \text{numero de filas} //$

F1



por cada columna se incrementa " * "

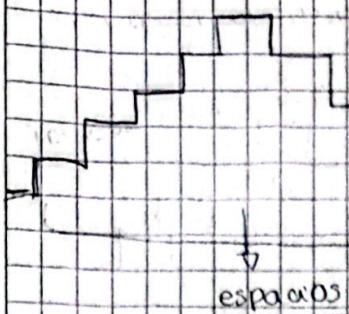
for ($i=1, i \leq n, i++$) {

 for ($j=1, j \leq i, j++$) {

 imprimir (" * ")

EST

T9)



5 nodos

secuencia: en cada nudo

T14)

1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 10 5 10 5 1

$$\text{nuevo valor} = \text{valor anterior} \times \frac{(j-i)}{(j+1)}$$

i → filas
→ columnas

88, 88, 88, 88, 88, 88, 88, 88, 88, 88

T19)

4
5
6
7
8
9
10
11
12
13
14
15
16

if $j=1 \rightarrow$ imprimir \rightarrow imprimir 60 60 60

$j=i \rightarrow$ imprimir 60

j está entre 2 y $i-1 \rightarrow$

imprimir numero
 \downarrow

aumenta
al cambiar de
fila

$j \rightarrow$ columnas
 \rightarrow filas

999999999 6666666 33333 666 333

Cadenas

-Casi Pedir una frase y presentarla invertida con las vocales en mayúsculas.

① Pedir al usuario la palabra → se usa el scanner. \rightarrow scanner

palabra = scanner.nextLine();

StringBuilder invertida = new StringBuilder(word).reverse();

genera una nueva cadena, que es la palabra original al revés.

String Builder = new StringBuilder(); construir la cadena final.

for → recorre cada carácter de la palabra invertida.

char letra = invertida.charAt(i) → obtiene el carácter en la posición i

if (Character.isLetter(indexOr(letra)) == true) → verifica si es una vocal, sea en minúscula o mayúscula.

Character.toUpperCase(char) → convierte en mayúsculas.

Arreglos

A01) Crear un array para cada palabra, de su nombre e ingrese el porcentaje de agua para cada palabra.

① Almacenar el nombre completo

nombreCompleto.split → divide la cadena nombre completo en un arreglo de palabras que separa donde hay espacios en blanco.

palabras = ["Anahi", "Valentino", "Pillijo", "Telan"],

Arreglo para porcentajes

int porcentajes = new int [palabras.length]

-② Pedirle al usuario que ingrese los porcentajes

porcentajes[i] = scanner.nextInt(), lee el número ingresado por el usuario ~~100~~ y lo guarda.

int barra = porcentaje / 10 → cuantos barras se dibuja

print if → para imprimir lo que se necesita

Loading

L01) Indicada de carga de 0 al 100%. Usar los signos \|\| para simular un movimiento rotación de carga de 0 hasta 100%.

① Crear un arreglo para los símbolos.

String[] rotacion = {"|", "\\", "/", "-"}

int carga = 0 → inicializa en 0,

ESTI

bucle para la carga

while (carga <= 100)

 // → mueve el cursor de la misma linea

 rotacion [carga % 4] → accede al arreglo {carga aumenta forma valores
 de 0 a 3}

carga += 5 → imprime los valores de 5 en 5

Thread.sleep(200); pausa la ejecución para crear el efecto de la animación

→
L06) Crear una barra de 20 cuadros, la barra \Rightarrow se desplaza de izquierda a derecha.

- (1) definir longitud de la barra:
- (2) variable para el progreso.

int posicion = (progreso * longitud de barra) / 100;

bucle para imprimir la barra.

```
for (int i=0; i < longitud barra; i++) {  
    if (p == posicion) {  
        imprimir ("=");  
    } else { imprimir (" ") }  
}
```

progreso += 5 → incrementa de 5 en 5

Thread.sleep(200)

+
se encarga de la animación

L07) Generar un número aleatorio este número debe servir para dibujar la señal de forma simétrica izquierda, derecha. Agregar un color.

—1— → nivel 2 // → ejemplo //

- (1) Utiliza Random para crear el número

int nivel = número aleatorio

String builder = new String builder (); generar cadenas que cambian.
append() permite añadir al final de una cadena existente).

Para imprimir (\Rightarrow) se necesitarán dos bucles.

(1) → se añade con appended //

codigo del color =

⇒ poner en la impresión //

Revisión

P4) Crear un método recursivo para calcular la potencia a^b .

① Pedir al usuario que ingrese la base int a

② Pedir que ingrese el exponente, int b

③ Mostrar el resultado.

Calcular el resultado

Si $b = 0 \rightarrow$ devolver 1 //

Método recursivo

a \neq calcular $(a, b-1)$,

2³

a = 2

b = 3

resultado //

2 calcula $(2, 2)$ $2 \times 4 = 8 //$

2 calcular $(2, 1)$ $2 \times 2 = 4$

2 calcular $(2, 0)$ como $b = 0 \rightarrow$ devolver 1 // $2 \times 1 = 2$

} llamadas

Automatas

A03) Crear un automata que valide $a+b+c$

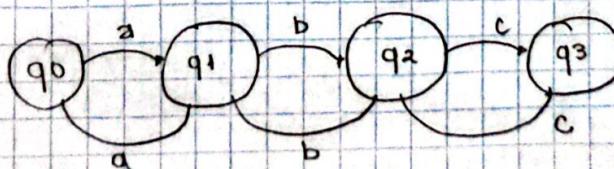
$a+b+c$ } significa que al menos una vez deben estar para ser válidas

$$Q = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{a, b, c\}$$

$$T = \{q_1, q_2, q_3\}$$

$$L = \{abc, aabc, abbc, \\ abcac, \dots\}$$

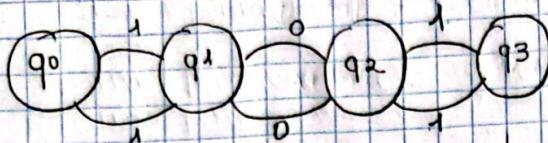


Matriz de transición

Σ	a	b	c
Q	1	2	3
q_0	1	e	e
q_1	1	2	e
q_2	e	2	3
q_3	e	e	1

A03 / Crear un automata que valide

$1 + 10 + 1^*$.



Matriz de transición

Σ	1	0
Q		
q0	1	e
q1	e	2
q2	3	2
q3	3	e

$$Q = \{q_0, q_1, q_2, q_3\}$$

$$\Sigma = \{1, 0\}$$

$$F = \{q_1, q_2, q_3\}$$

$$L = \{10, 1100, 1100111\dots\}$$

Más ejercicios

F8).



→ cuando fila = 0

columna = nivel → que ingresa el usuario.

fila > 0

columna = nivel - fila.

↓
cuando no se cumple

a " " → espacio

C4). Pedir una frase y eliminar la letra pedida de la frase.

① Pedir al usuario una frase → arreglo.

② Ingresa una letra para eliminar

frase. Peloce (letra, "") → eliminar la letra.

} replace() → crea una nueva cadena donde la letra se reemplaza con ""

resultado =

Q5). Crear una barra es de 20 caracteres, la barra va cambiando la punta.

① Definir longitud de la barra. = 20
char [punta] = {>, - }

int progreso = 0 → inicializar en 0.
int punto indice = 0

while (progreso <= 100) se ejecuta si es menor igual a 100.

calcular cuantos "—" se deben presentar

-(progreso * longitud de barra) / 100.

String Barra1 = " "

for = (int i = 0; i < progreso; i++) {

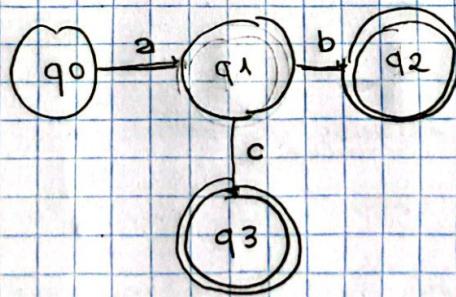
{ barra.append ("-") ;

agregar la punta de la barra. (efecto de animación.).

progreso = progreso + 5 (que raya de 5 en 5).

Thread.sleep (para crear animación).

A02) Crear un automata que valide ab + ac



$$\Sigma = \{a, b, c\}$$

$$Q = \{q_0, q_1, q_2, q_3\}$$

$$F = \{q_3\}$$

→ Matriz de transición

	a	b	c
q			
q ₀	1	e	e
q ₁	e	2	3
q ₂	e	e	e
q ₃	e	e	e

L = {ab, ac} | l

e = error.