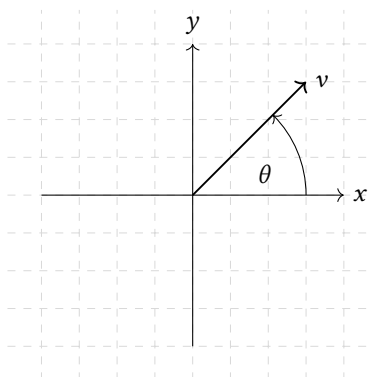


1. Dibujar un diagrama de flujo para los siguientes problemas.
 - (a) Un programa que imprime el valor absoluto de un número entero a que acepta como argumento.
 - (b) Un programa que imprima el máximo de tres argumentos enteros a , b y c .
 - (c) Un programa acepta como argumento un número entero y decide si es positivo, negativo o cero.
 - (d) Un programa que imprime las raíces de una función cuadrática. Primero se debe calcular el valor del discriminante $D = b^2 - 4ac$. Si $D > 0$ entonces se imprimen dos raíces. Si $D = 0$ entonces la raíz es doble y se imprime una sola vez. En el caso de que $D < 0$ se informa que no existe solución en los números reales.
 - (e) Se tira un dado “cargado” donde la probabilidad de sacar un 6 es de $3/8$ y para las otras 5 caras de $1/8$. El programa imprime el resultado de tirar el dado.
 - (f) Un programa acepta tres argumentos, medio de pago (un char que puede ser 'e', 'd', 't' o 'c'), precio y cantidad. Calcular e imprimir el monto total a pagar. Si es en efectivo aplicar un 15% de descuento, por transferencia un 10%. Con tarjeta de crédito recargar un 5% y no hacer nada con tarjeta de débito.
 - (g) Un programa suma los primeros n números naturales. Al final imprime el valor de la suma.
 - (h) Un programa imprime una tabla con los números enteros positivos y su cuadrado, hasta un número n leído como argumento.
2. Codificar en C los programas del ejercicio anterior.
3. Escribir un programa en C que use la función `atan2()` de `math.h` para calcular el ángulo que forma un vector de coordenadas (x, y) con el eje x . Por ejemplo el vector $(1, 1)$ forma un ángulo de 45° y el vector $(0, -1)$ un ángulo de 270° . La función `atan2(y, x)` calcula la arcotangente de y dividido x .



4. Implementar en C el programa que calcule el factorial de n , siendo n un entero mayor o igual a cero. Si el argumento n ingresado por línea de comandos es menor a cero imprimir “Error”. Dar el diagrama de flujo del programa.

```
$ ./factorial 4
24
$ ./factorial 0
1
$ ./factorial -2
Error
```

5. Escribir un programa que acepte tres argumentos enteros e imprima “iguales” si los tres números son iguales o “no iguales” de lo contrario. Dar el diagrama de flujo.
6. Escribir un programa que acepte un argumento entero n , la cantidad de veces que el programa imprime “Hola mundo”. Dar el diagrama de flujo.

7. Escribir un programa que imprima los números entre el 1000 y el 1999, mostrando 5 números por línea. Dar el diagrama de flujo.
8. Escribir un programa que acepte un argumento entero n e imprima n números aleatorios entre 0 y 100. Al finalizar el programa debe imprimir el promedio de los números que aparecieron. Dar el diagrama de flujo.
9. Escribir un programa que use un ciclo `for` para imprimir los primeros n números de Fibonacci, siendo n un argumento entero del programa. La sucesión de Fibonacci está definida de la siguiente manera.

$$F_0 = 0, F_1 = 1$$

Es decir los dos primeros números son cero y uno. A partir del tercer número (F_2) y en adelante cada número es la suma de los dos anteriores.

$$F_n = F_{n-2} + F_{n-1}$$

Por ejemplo, los primeros diez números de Fibonacci $F_0 \dots F_9$ son:

$$0, 1, 1, 2, 3, 5, 8, 13, 21, 34$$

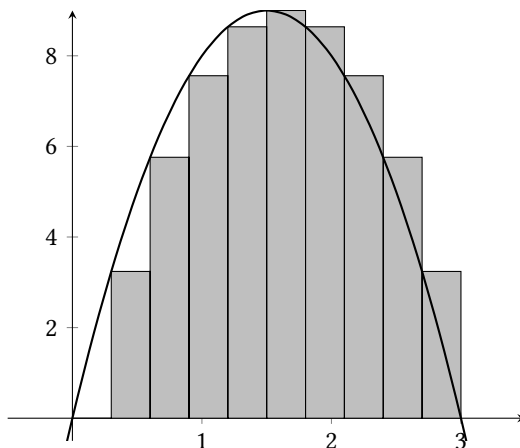
10. Usando el algoritmo de Euclides para encontrar el MCD (máximo común divisor) escribir un programa que acepte como argumentos dos números enteros a y b e imprima su MCD. El algoritmo de Euclides en pseudocódigo sería así:

```

mientras  $b \neq 0$  hacer
     $t \leftarrow b$ 
     $b \leftarrow a \bmod b$ 
     $a \leftarrow t$ 
fin
imprimir  $a$ 
    
```

El valor final de a es el resultado del algoritmo, es decir el MCD de a y b . Codificar el programa y dibujar el diagrama de flujo de acuerdo al pseudocódigo de arriba.

11. Escribir un programa que acepte dos argumentos enteros a y b con $b > a$. El programa debe imprimir todos los números divisibles por 3 en el intervalo $[a, b]$. Si a es mayor a b el programa debe imprimir un mensaje indicando al usuario el uso correcto del programa y salir. Dibujar el diagrama de flujo.
12. Escribir un programa que acepte como argumento un número entero e imprima el número con los dígitos al revés. Por ejemplo para la entrada 12345 debe imprimir 54321.
13. Escribir un programa que decida si un entero positivo n es primo o no. Un número es primo si es divisible sólo por uno y por sí mismo. Por ejemplo 7 es primo pero 33 no. Dar el diagrama de flujo.
14. Se quiere calcular el área debajo de la parábola delimitada por la función $f(x) = -4x^2 + 12x$ y el eje x en el intervalo $[0, 3]$.



Para eso se debe escribir un programa que divida el intervalo $[0, 3]$ en n partes iguales, siendo n un argumento entero del programa. Para aproximar el área se calculan las áreas de n rectángulos de igual base y de altura $f(x)$ evaluando la función del modo que se ilustra en el gráfico. El área total debajo de la función se puede aproximar sumando las áreas de todos los rectángulos. Eligiendo un n más grande se obtiene una mejor aproximación. El programa debe imprimir el área total con dos lugares después de la coma como se indica debajo. Si el usuario ingresa $n < 1$ el programa imprime "Error".

```
$ ./area 100
18.00
$ ./area 0
Error
```

Implementar el programa en C y dar el diagrama de flujo del mismo.

15. Dar el diagrama de flujo para un programa que imprime la matriz identidad de $n \times n$. Usar dos *loops* anidados con variables de control i y j . Implementar el programa en C, siendo n un entero ingresado por línea de comandos.
16. Usar dos ciclos *for* anidados para imprimir un patrón en forma de tabla con asteriscos que indique si i divide a j o j divide a i . Usar un argumento entero n para el tamaño de la tabla. Ejemplos:

```
$ ./divisores 4
1 2 3 4
* * * * 1
* *   * 2
*  *   3
* *   * 4
$ ./divisores 6
1 2 3 4 5 6
* * * * * 1
* *   *   2
*  *     * 3
* *   *   4
*       *  5
* * *     * 6
```

Escribir el programa y dar el diagrama de flujo.

17. Escribir un programa en C que imprima una tabla con los valores de las funciones: $\ln n$, n , $n \ln n$, n^2 , n^3 , 2^n para los valores de n : 1, 2, 4, 8, 16, 32, 64. Usar caracteres `\t` para alinear las columnas y las funciones `pow()` y `log()` de `math.h`. Dar el diagrama de flujo.
18. Escribir un programa en C que acepte un número arbitrario de argumentos enteros por línea de comandos e imprima el máximo de los números ingresados.

```
$ ./args_maximo 29 23 14 2984 999 1000
2984
$ ./args_maximo 6 3 5 10 7 2
10
```

Dar el diagrama de flujo.

19. Escribir un programa como el anterior pero que también reporte el mínimo y el promedio de los números ingresados.

20. Escribir un programa que acepte un argumento entero n y use dos *loops* anidados para imprimir un patrón como el de un tablero de ajedrez usando asteriscos y espacios en la terminal. El argumento n es la cantidad de filas y columnas del tablero.

```
$ ./ajedrez 3
* *
 *
* *
$ ./ajedrez 8
* * * *
 * * * *
* * * *
 * * * *
* * * *
 * * * *
* * * *
 * * * *
* * * *
```

21. Escribir un programa que acepte un argumento entero n e imprima los números de 1 hasta n y su cuadrado. Por ejemplo:

```
$ ./cuadrados 3
1 → 1
2 → 4
3 → 9
$ ./cuadrados 5
1 → 1
2 → 4
3 → 9
4 → 16
5 → 25
```

Implementar el programa en C y dibujar el diagrama de flujo.

22. Escribir un programa que imprima los números en el intervalo del 1 al 100 que al ser divididos por un argumento entero n , su resto sea igual a 3. Dibujar el diagrama de flujo.
23. Escribir un programa que acepte dos argumentos enteros x , y representando coordenadas en el plano cartesiano. El programa debe imprimir a qué cuadrante pertenece el punto. Los cuadrantes del plano cartesiano se representan con los números romanos I, II, III y IV.
24. Escribir un programa que acepte dos números enteros a y b y calcule su cociente. Si la división de los números no tiene resto igual a cero entonces imprimir “No se pueden dividir” y salir del programa. Por ejemplo, $a = 20$, $b = 5$ debe imprimir “4”. Con $a = 20$ y $b = 3$ debe imprimir “No se pueden dividir”.
25. Dibujar el diagrama de flujo para un programa que acepta un argumento entero $n \geq 2$ e imprime todos los números $x \in \mathbb{N}$ tal que x es par hasta n inclusive. Implementar el programa en C.
26. El problema de Basilea consiste en encontrar la suma exacta de los recíprocos de los cuadrados de los enteros positivos. Es decir el resultado de la siguiente serie infinita:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{1}{1^2} + \frac{1}{2^2} + \frac{1}{3^2} + \cdots + \frac{1}{n^2}$$

Es sabido gracias a Euler que el resultado de la suma es:

$$\sum_{n=1}^{\infty} \frac{1}{n^2} = \frac{\pi^2}{6}$$

Usar esta información para implementar en C un programa que aproxime correctamente el número π hasta 5 dígitos después de la coma. Dibujar el diagrama de flujo.

27. Escribir un programa que acepte un argumento entero n e imprima las potencias positivas de dos hasta 2^n , una por línea. Dibujar el diagrama de flujo.
28. Escribir un programa que acepte un entero positivo n como argumento e imprima el mismo número en binario. Para realizar la conversión tener en cuenta que un número en binario está expresado como suma de potencias de dos. Primero buscar la potencia de dos más grande que entra en el número a convertir. Ese es el uno que está más a la izquierda. Luego ir probando con las potencias de dos sucesivas en orden decreciente. Si sumamos una potencia y no nos pasamos del número entonces va un uno, si nos pasamos va un cero. Repitiendo estos pasos obtenemos la representación de n en binario. Dar el diagrama de flujo.

29. Escribir un programa que imprima el n -ésimo número armónico. Es decir la suma de los recíprocos de los primeros n números naturales.

$$H_n = \sum_{k=1}^n \frac{1}{k} = 1 + \frac{1}{2} + \frac{1}{3} + \dots + \frac{1}{n}$$

30. Un apostador juega a cara o ceca apostando siempre \$1,00 (un peso) por apuesta. Cuando gana duplica su apuesta y cuando pierde, pierde lo que apostó. El apostador comienza con una cantidad de dinero s y decide dejar de apostar cuando llega a una cantidad g de dinero. O cuando pierde todo su dinero. Usar un programa para simular una cantidad t de experimentos y determinar la probabilidad de que el apostador gane (llegue a su objetivo) expresada en porcentaje. También reportar la cantidad promedio de apuestas por experimento. Los argumentos del programa son s , g y t en ese orden, todos enteros. Ejemplos:

```
$ ./apostador 500 2500 100
23% de victorias
Promedio de apuestas: 1113791
$ ./apostador 10 20 1000
51% de victorias
Promedio de apuestas: 98
```

Dibujar el diagrama de flujo del programa.

31. Escribir un programa que realice la factorización de enteros en factores primos. Ejemplos:

```
$ ./factores 60
2 2 3 5
$ ./factores 72
2 2 2 3 3
```

Por el teorema fundamental de la aritmética todo número natural tiene una descomposición única en factores primos. Por ejemplo:

$$72 = 2 \cdot 2 \cdot 2 \cdot 3 \cdot 3$$

Implementar el programa en C y dibujar el diagrama de flujo.

32. Un grupo infinito de matemáticos entra a un bar. El primero le pide al cantinero una pinta de cerveza, el segundo media pinta, el tercero un cuarto de pinta, el cuarto un octavo de pinta y así sucesivamente. El cantinero los mira, sirve dos pintas y les dice “Arréglense ustedes”.

Escribir un programa que demuestre por qué el cantinero tenía razón al servir dos cervezas. Dibujar el diagrama de flujo.

33. En un concurso televisivo el conductor le presenta al participante tres puertas cerradas. Detrás de una de ellas hay un auto. En las otras dos hay cabras. El concursante debe elegir una puerta. Después de haber elegido una puerta el conductor abre una de las dos puertas restantes descubriendo una cabra y le ofrece al concursante la posibilidad de cambiar de puerta. ¿Debería cambiar de puerta o mantener su decisión original? ¿Da lo mismo?

Escribir un programa que simule jugar al juego n veces. Cambiando y sin cambiar de puerta. Imprimir la probabilidad de ganar el juego para cada una de las dos estrategias: siempre cambiar de puerta o siempre mantener su primera elección.

34. Escribir un programa que implemente el método de Newton para aproximar el valor de la raíz cuadrada de un número real c . El algoritmo es el siguiente.

```
t ← c
mientras |t - c/t| > ε · t hacer
  | t ← (c/t + t)/2
fin
imprimir t
```

Donde t es el estimado de \sqrt{c} y ε es el error. Para calcular la raíz cuadrada con una precisión de 15 dígitos después de la coma asignar a ε un valor de 10^{-15}

Implementar el programa en C y dibujar el diagrama de flujo.

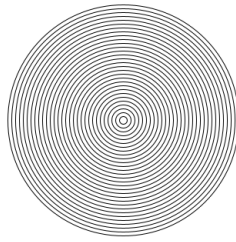
35. Escribir un programa que acepte un número arbitrario de argumentos. El programa calcula el producto de los números tal como aparecen en la línea de comandos hasta encontrar un cero o hasta que se terminan los argumentos. Luego del primer cero ignora el resto de los argumentos. Ejemplos

```
$ ./producto 2 2 2 2
16
$ ./producto 2 3 4 0 1234 12
24
```

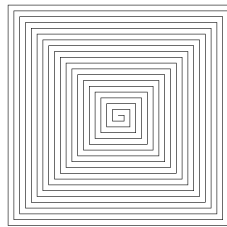
Implementar el programa en C y dar el diagrama de flujo.

36. Escribir un programa en C que reciba dos argumentos enteros x e y y declare dos variables con los mismos nombres. El programa tiene que asignar a x el menor de los dos valores, independientemente del orden en el que fueron ingresados. Dar el diagrama de flujo.
37. Escribir un programa que simule tirar una moneda. Imprimir “Cara” o “Ceca”. Dibujar el diagrama de flujo.
38. Escribir un programa que imprima el abecedario del español en una sola línea separando cada letra con un espacio. Dar el diagrama de flujo.
39. Escribir un programa que acepte dos números enteros positivos a y b y decida si los números son capicúas, es decir que se leen igual de izquierda a derecha que de derecha a izquierda. Dar el diagrama de flujo.

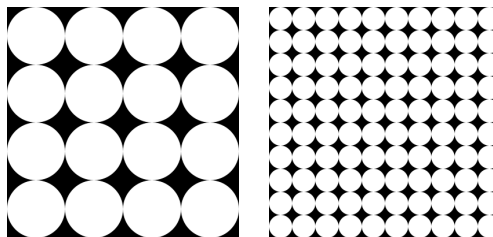
40. Escribir un programa en Processing que dibuje círculos concéntricos en una ventana de 400×400 píxeles. Usar la función `ellipse()`. La imagen debajo ilustra el resultado aproximado.



41. Escribir un programa en Processing que dibuje una espiral “cuadrada” como la que se ilustra. Usar la función `line()`.



42. Escribir un programa en Processing que acepte un argumento entero n y dibuje $n \times n$ círculos blancos sobre un fondo negro. El tamaño de la imagen es 400×400 píxeles. El argumento n tiene que cumplir $4 \leq n \leq 100$ y n debe ser divisor de 400.



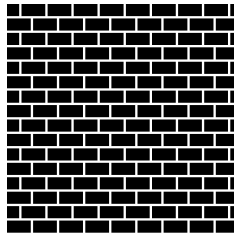
Izquierda $n = 4$, derecha $n = 10$

43. Escribir un programa en Processing que dibuje n círculos de radio aleatorio y de color aleatorio en un lienzo de 400×400 píxeles. El radio r de cada círculo debe cumplir $10 \leq r \leq 60$. La ubicación de cada círculo en el lienzo también debe ser aleatoria, sin ninguna restricción.



Ejemplos con $n = 20$, $n = 30$ y $n = 50$

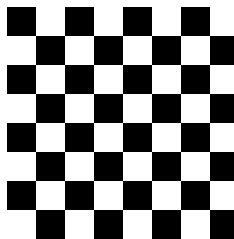
44. Dibujar en Processing un patrón como de ladrillos en una pared de 400×400 píxeles como se ilustra abajo.



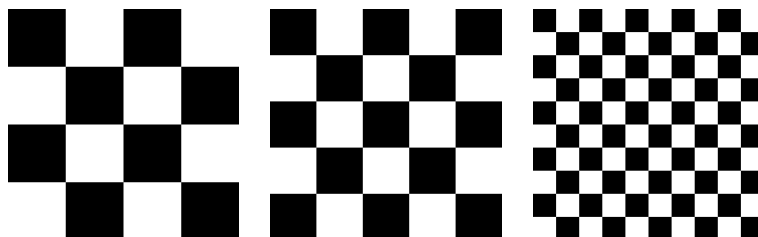
45. Escribir un programa en Processing que dibuje una línea como se ilustra debajo. El proceso debe ser aleatorio. Comenzando desde la izquierda al medio de la ventana dibujar una línea a 45° hacia abajo de longitud aleatoria. Luego girar 90° y dibujar una línea hacia arriba también de longitud aleatoria y repetir desde el principio. En el ejemplo el lienzo es de 600×200 píxeles.



46. Escribir un programa en Processing que dibuje un tablero de ajedrez: 64 casilleros, 8 filas y 8 columnas, alternando negro y blanco. Debe quedar como se ilustra abajo. El tamaño de la imagen debe ser un cuadrado de 400 píxeles de lado.

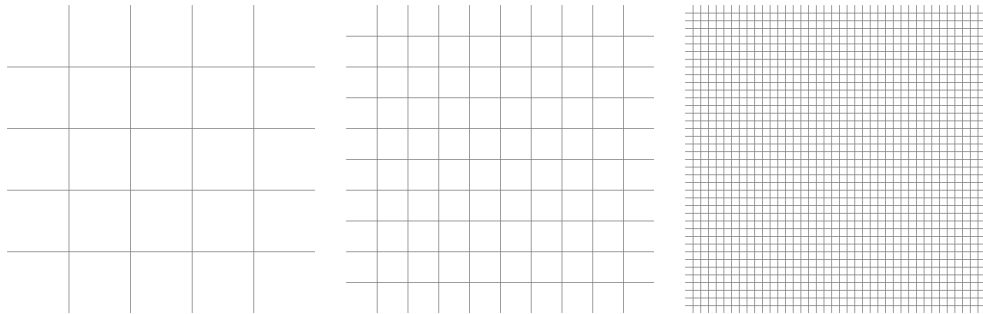


47. Escribir un programa en Processing como el anterior pero que dibuje un tablero de $n \times n$ casilleros alternando blanco y negro. Usar un argumento entero para n .



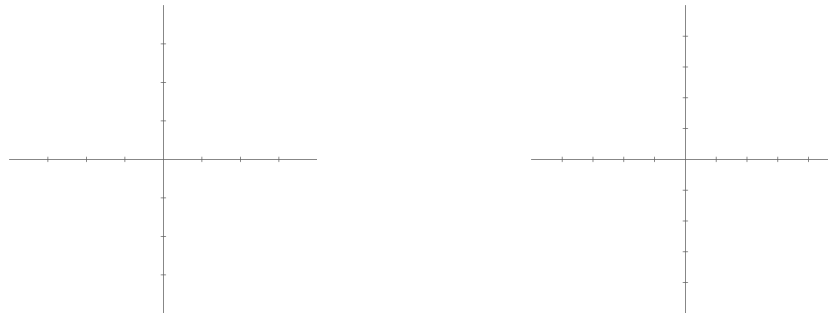
Ejemplos con $n = 4$, $n = 5$ y $n = 10$

48. Escribir un programa en Processing que acepte un argumento entero n y dibuje una grilla de $n \times n$ casilleros en un lienzo de 400×400 píxeles.



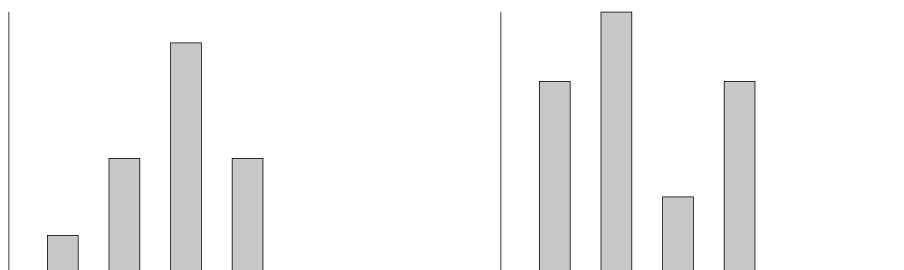
Ejemplos con $n = 5$, $n = 10$ y $n = 40$

49. Escribir un programa en Processing que acepte un argumento entero n . El programa dibuja los dos ejes del plano cartesiano marcando n divisiones en el eje x y en el eje y como se ilustra debajo.



$n = 8$ y $n = 10$

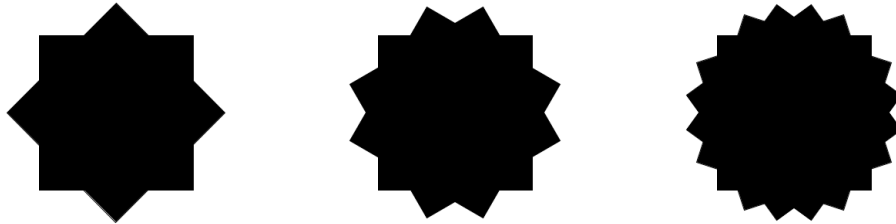
50. Escribir un programa en Processing que acepte cuatro argumentos enteros a , b , c y d . La cantidad de habitantes en cada franja etaria: 0 - 12, 13 - 21, 22 - 60 y 60 en adelante, respectivamente. El programa utiliza los cuatro datos para dibujar un histograma como se ilustra abajo.



$\{a, b, c, d\} = \{50, 150, 300, 150\}$

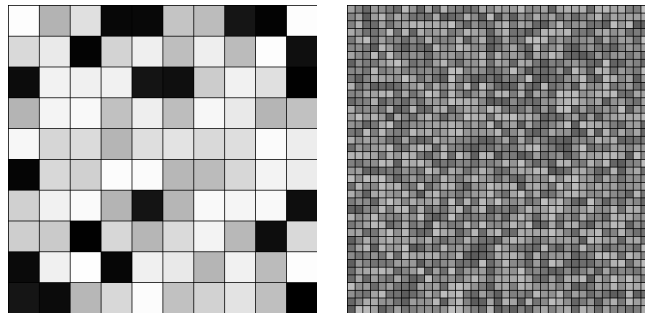
$\{a, b, c, d\} = \{250, 340, 100, 250\}$

51. Escribir un programa en Processing que acepte un único argumento entero n , la cantidad de puntas de una “estrella”. El programa debe dibujar una estrella con n puntas siempre y cuando $n \geq 4$ y n sea múltiplo de 4. Si alguna de esas condiciones no se cumple el programa termina usando la función `exit()`. Las estrellas se deben dibujar usando cuadrados con `rect()` e ir rotándolos. Investigar en la referencia de Processing las funciones `rotate()` y `translate()`. El tamaño de la ventana es de 400×400 píxeles.



Ejemplos con $n = 8$, $n = 12$ y $n = 20$

52. Escribir un programa en Processing que acepte un argumento entero n y divida un lienzo de 400×400 píxeles en $n \times n$ casilleros. En cada casillero dibujar un cuadrado con `rect()` usando un color aleatorio.



Ejemplos con $n = 10$ y $n = 40$