

## Entrega

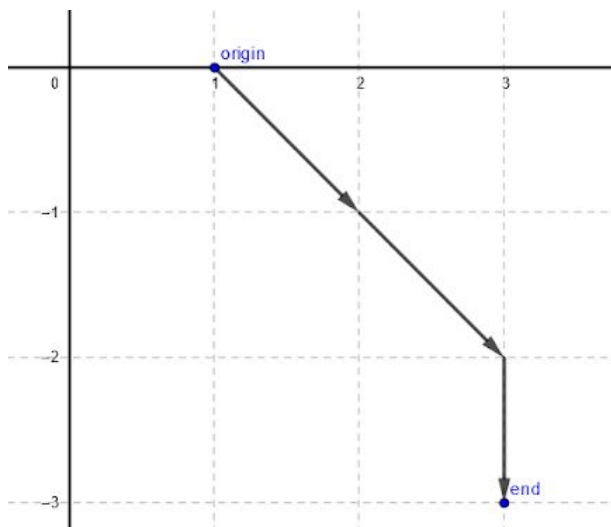
Las soluciones se entregarán en **un único archivo .java**, donde estarán todos los métodos solicitados. En el main, se pueden poner casos de pruebas de los métodos implementados.

## Consideraciones

- Recuerda que para hacer media, se tiene que obtener el **30% como mínimo**, de la puntuación de cada ejercicio.
- Para los métodos que solicita el ejercicio es necesario entre comentarios, **indicar**:
  - Precondiciones
  - Postcondiciones
  - Entrada
  - Salida
- Para el resto de métodos **auxiliares**, solo es necesario indicar una **descripción**.

### Ejercicio 1 (4,5 puntos)

Crea un método llamado **int printPath(int xOrigin, int yOrigin, int xEnd, int yEnd)**, el cual tomando las coordenadas cartesianas de 2 puntos, trace el camino más corto (el que requiere menos pasos) empezando por la coordenada origin, y terminando en la coordenada end, representadas por (xOrigin,yOrigin) y (xEnd,yEnd). Teniendo en cuenta que en cada paso es posible aumentar o decrementar la x, la y, o ambas a la vez. El método devolverá el número de pasos tomados.



Así, por ejemplo, para  
`printPath (1, 0, 3, -3)`, se  
trazaría el siguiente camino:

Origin: (1,0)  
(2,-1)  
(3,-2)  
End: (3,-3)

**y devolvería: 3**

### Ejercicio 2 (3 puntos)

Crea un método llamado **printParallelogram(int altura, int base, boolean printAsteriscos)** para dibujar un paralelogramo, el cual tendrá como parámetros: Un entero para indicar la altura del paralelogramo, un entero para indicar la base y un booleano que indique si se muestran asteriscos (true), o números.

El paralelogramo que se dibuje, en caso de dibujarse con asterisco, tendrá a su derecha una línea discontinua. El paralelogramo deberá tener la siguiente forma:

```
****|
****|
****|
****|
****|
```

(Ejemplo para 5, 4, true)  
(Se dibuja con asteriscos y con la línea a la derecha)

```
*****|
*****|
*****|
```

(Ejemplo para 3, 6, true)  
(Se dibuja con asteriscos y con la línea a la derecha)

```
444444
333333
222222
111111
```

(Ejemplo para 4, 6, false)  
(Se dibuja con números y sin línea a la derecha)

```
555555
444444
333333
222222
111111
```

(Ejemplo para el número 5, 6, false)  
(Se dibuja con números y sin línea a la derecha)

### Ejercicio 3 (1,5 puntos)

Observa el siguiente método y responde:

1. ¿Qué **tipo** de recursividad se realiza?
2. ¿**Cuántos** casos base y casos generales hay y **cuáles** son?
3. **Explica**, mediante comentarios, el proceso que realiza el método. Escribe un método iterativo equivalente en Java a `proc1`. Recuerda que todos los elementos de tu código deben utilizar nombres autodocumentados.

```
public static void proc1(int n1) {
    if(n1==1)
        System.out.println(n1+".");
    else {
        System.out.print(n1+",");
        proc1(n1-1);
    }
}
```