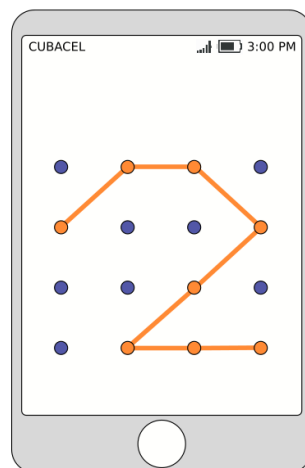


Patrones de Desbloqueo

NOTA: Si usted está leyendo este documento sin haber extraído el compactado que se le entregó, ciérrelo ahora, extraiga todos los archivos en el escritorio, y siga trabajando desde ahí. Es un error común trabajar en la solución dentro del compactado, lo cual provoca que los cambios no se guarden. Si usted comete este error y entrega una solución vacía, no tendrá oportunidad de reclamar.

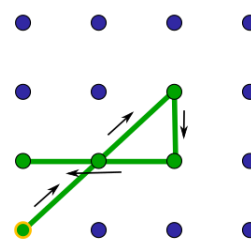
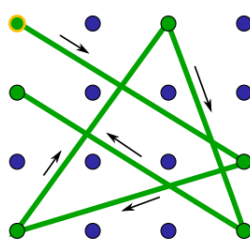
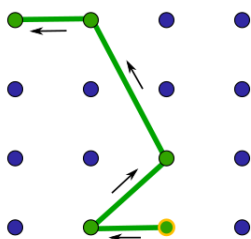
Usted ha sido contratado para evaluar la seguridad de una nueva gama de dispositivos móviles Android. Estos dispositivos presentan un patrón de desbloqueo similar al mostrado en la siguiente figura.



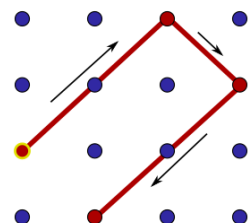
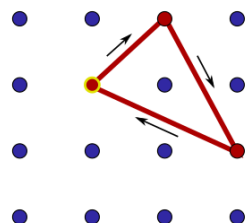
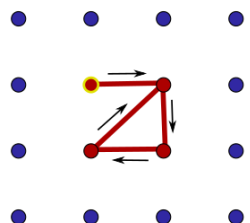
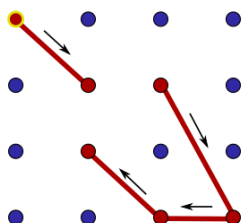
Un patrón de desbloqueo de tamaño k es trazado por un usuario con su dedo, conectando $k+1$ puntos de una matriz de n filas y m columnas igualmente espaciadas. Un patrón válido se define como:

- Una sucesión de puntos conectados que se considerarán conectados en el orden en que aparecen en la sucesión.
- Se puede pasar sobre un punto que ya está en el patrón pero no terminar en él, de modo que el punto inicial y final no pueden coincidir.
- Todo punto que sea intersectado por uno de los segmentos forma parte del patrón.

A continuación se muestran algunos ejemplos de patrones válidos e inválidos, junto a su explicación. Aparece resaltado con borde amarillo el punto inicial.



El patrón de la izquierda es válido trivialmente pues cumple con todos los requisitos, tiene tamaño **4**. El patrón del centro también es válido, pues aunque se conectan puntos que están bien alejados en la matriz, nunca se pasa un segmento por encima de un punto que no pertenezca al patrón, tiene tamaño **5**. El patrón de la derecha también es válido, pues aunque se pasa por el segundo punto dos veces la segunda vez es para cruzar sobre él, en este caso tiene tamaño **4** porque ese segundo punto solo se cuenta una vez.



El patrón más a la izquierda es incorrecto pues no está conectado. El segundo patrón es incorrecto pues el segundo punto se intenta reusar dos veces en el patrón (la segunda vez como punto final). El tercer patrón es incorrecto pues el punto final e inicial coinciden. El cuarto patrón es incorrecto pues se intenta conectar de un solo trazo 2 puntos alejados, pasando directamente por encima de otro punto que no se desea que pertenezca al patrón, y esto no es posible.

Usted debe haber recibido junto a este documento una solución de Visual Studio con dos proyectos: una biblioteca de clases (*Class Library*) y una aplicación de consola (*Console Application*). Deberá implementar el método `CantidadValidos` que se encuentra en la clase `Patrones` en el `namespace Weboo.Examen`. En la biblioteca de clases encontrará la siguiente definición:

```
namespace Weboo.Examen
{
    public class Patrones
    {
        public static int CantidadValidos(int k, int filas, int columnas)
        {
            //Borre la siguiente línea y escriba su código
            throw new NotImplementedException();
        }
    }
}
```

El método debe devolver la cantidad de patrones válidos que se pueden formar con **k** puntos. Se garantiza que **k** es mayor o igual que **1**, y los parámetros **filas** y **columnas** son mayores o iguales que **2**, pero pueden

ser diferentes entre sí. Es posible que para un cierto **k** no exista ningún patrón válido, en este caso usted debe devolver **0**. Por tanto, su código nunca debe lanzar excepción.

NOTA: Tenga en cuenta que 2 patrones que tengan la misma forma, pero trazados en un orden distinto, se consideran **diferentes**.

NOTA: Todo el código de la solución debe estar en este proyecto (biblioteca de clases), pues es el único código que será evaluado. Usted puede adicionar todo el código que considere necesario, pero no puede cambiar los nombres del namespace, clase o método mostrados. De lo contrario, el probador automático fallará. En particular, es imprescindible que usted no cambie los parámetros del método `CantidadValidos`, ni su orden. Por supuesto, usted puede (y debe) adicionar todo el código que necesite.

NOTA: Los casos de prueba que aparecen en este proyecto son solamente de ejemplo. Que usted obtenga resultados correctos con estos casos no es garantía de que su solución sea correcta y de buenos resultados con otros ejemplos. De modo que usted debe probar con todos los casos que considere convenientes para comprobar la validez de su implementación.