

Primer examen intrasemestral de Programación
Curso 2024-2025
Parte 2

Jaime y la piña

El del vínculo, Jaime el del vínculo, tenía una misión muy importante. Debía entrar al almacén de un hombre muy rico que el gobierno había tachado como “mala gente” y “decomisar” sus recursos. *¿Por qué el hombre era mala gente? ¿Qué había hecho de malo? ¿Si él era mala gente, entonces el gobierno estaba formado por “los buenos”? ¿Entonces quién decide quién es bueno y quién es malo? ¿Era correcto lo que estaba haciendo?* Por suerte, Jaime no era filósofo sino un héroe nacional. Como todo buen héroe reconocido por la nación, le bastaba con hacer lo que los de arriba le ordenaban.

Luego del allanamiento legal y de poner a dormir a varios guardias “mala gente”, llega a la puerta del almacén y procede a abrir el candado. El candado, para sorpresa de Jaime, tenía la forma de una piña (el hombre rico tenía gustos raros). La piña tenía n filas que le daban la vuelta de forma circular con m columnas. Cada una de las $n * m$ casillas tenía un número entero asociado y las filas rotaban individualmente hacia la izquierda o hacia la derecha. También había una marca en una de las columnas, a la que llamaremos **columna central**.



Afortunadamente, los altos mandos del gobierno le habían dado a Jaime la combinación que abriría el candado. Desafortunadamente, aún teniendo la combinación, el proceso de apertura no era nada sencillo. La combinación constaba de dos listas (arrays) de números enteros.

La primera lista, que llamaremos *factors*, tenía tamaño n y representaba el factor de movimiento de cada una de las filas de la piña. Por ejemplo, si en la posición i de *factors* había un 5, significaba que la fila i de la piña debía ser girada hacia la derecha de 5 casillas en 5 casillas. Si había un -3, entonces debía girarse hacia la izquierda de 3 casillas en 3 casillas. Si había un 0, esa fila no debía girarse.

La segunda lista tenía, *operations* un tamaño arbitrario k . Esta representaba la cantidad de operaciones que se debía hacer para abrir el candado. Por ejemplo, si la lista fuera la siguiente $[1, 3, -2]$, entonces la fila de factor 5 debía girarse 5 veces a la derecha, luego girarse 15 veces a la derecha y por último 10 veces a la izquierda.

Para desbloquear el candado, se debe obtener un número entero. Este número es la suma de todos los números que van quedando en la **columna central** luego de realizada cada operación. Sin embargo, las casillas que ya fueron sumadas, no pueden volver a sumarse. Ayude a Jaime a decomisar los recursos ilícitos del hombre “mala gente” cuyo nombre no vale la pena mencionar.



Su tarea será implementar el siguiente método en el archivo *exam/Solution.cs*:

```
1  public static int Solve(int[,] matrix, int[] factors, int[]  
    operations)  
2  {  
3      throw new NotImplementedException();  
4  }
```

Se recibe como entrada la matriz que representa la pñña, el array de los factores y el array de las operaciones. Debe retornar el número entero conseguido después de realizar todas las operaciones de *operations*. La **columna central** estará formada por la columna $\frac{m}{2}$. Se garantiza que m siempre será impar y mayor o igual que 3.

En el archivo *Program.cs* se muestran algunos ejemplos, sin embargo, usted puede (y debe) crear sus propios casos para poner a prueba su solución. A continuación se explican los ejemplos dados:

Ejemplo

factors

5-20

operations

13-20

123123123

5-20

231312123

+6

5-20

231312123

+0

5-20

312231123

+4

5-20

312231123

+0

=10