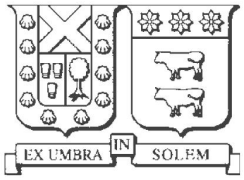
Departamento de Informática
Universidad Técnica Federico Santa María



INF-134: Estructura de Datos 2018-2
Tarea 1B

John Rodríguez Mora
jirodrig@alumnos.inf.utfsm.cl

Hubbert Hoffmann
hoffmann@inf.utfsm.cl

1. Problemas

En la presente tarea deberán programar dos algoritmos que dan solución a dos problemas de arreglos y matrices.

- **¿Existen dos enteros que sumen x?**

  Sean 2 arreglos A y B de números enteros, de tamaños n y m respectivamente, y un número entero x. Se quiere determinar si existen 2 números $a \in A$ y $b \in B$ tal que satisfacen la ecuación $a + b = x$. Para esto, el programa debe seguir los siguientes pasos:

  - Ingresar dos enteros positivos correspondiente a n y m.
  - Ingresar lo enteros del arreglo A, en una misma línea, separados por espacios.
  - Ingresar lo enteros del arreglo B, en una misma línea, separados por espacios.
  - Ingresar el resultado que debe dar la suma, correspondiente al entero x.
  - Si existen 2 enteros, uno perteneciente a cada arreglo, que sumen x, se deben imprimir ambos en una misma linea, separados por un espacio. En caso contrario imprimir `no existen`.

- **El piso es lava!**

  El ayudante John va tranquilamente caminando cuando el profesor Hoffmann grita **el piso es lava!**. John tiene ciertas superficies alrededor donde puede pararse y salvarse de la lava, pero debe poder llegar a ellas con un salto o se quemará. Dada una matriz cuadrada de tamaño n que simboliza el piso, donde cada casilla puede ser 0, si es que hay lava, o 1 si es una superficie segura, y la posición (fila, columna) de John junto con la cantidad de casillas x que puede saltar (vertical, horizontal o diagonalmente), diga si es que puede salvarse o no.

  El programa sigue la siguiente estructura:

  - Ingresar n.
  - Ingresar elementos en la matriz.
  - Ingresar posición fila, columna donde esta John
  - Ingresar x.

- Mostrar `se salvo` si es que con un salto logra pararse en una superficie segura, o `se murio` en caso contrario.

Observación: las posiciones comienzan desde 0. Filas son aquellas horizontales, y columnas son verticales.

2. Ejemplos de entrada y salida

(Input en azul, output en rosado y posición de John en verde.)

Ejemplo 1

```
$ ./tarea1B
3 4
1 2 3
4 5 6 7
7
2 5
6
1 1 0 0 0 0
1 1 0 0 0 0
0 0 0 0 1 0
0 0 0 0 1 0
0 0 1 0 0 0
0 1 1 0 0 0
1 3
1
se salvo
```

Ejemplo 2

```
$ ./tarea1B
5 5
8 19 1 0 -3
7 2 5 -1 -10
11
no existen
5
1 1 0 0 0
1 1 0 0 0
0 0 0 0 1
0 0 0 0 1
0 0 0 0 0
4 0
2
se murio
```

En el ejemplo 1, se entrega 2 y 5 como resultado, pero también puede ser 3 y 4. John se salva en el ejemplo 1 pues la posición (1, 3) esta a una casilla de (2, 4), que es una superficie segura. En cambio, en el ejemplo 2 muere por que no hay superficies seguras en un radio de 2 casillas.

3. Consideraciones adicionales

- Los datos de entrada serán correctos, no debe realizar verificación.
- El largo del arreglo nunca será mayor a 100.
- La cantidad de filas y columnas de la matriz nunca serán más de 100 cada una.

4. Bonificaciones adicionales

+5 pts: Escribir en los supuestos del README la complejidad temporal de cada algoritmo, acompañado de una explicación.

5. Consideraciones de código

- Solo se puede usar arreglos básicos de C++.
- Debe seguir el formato de inputs y outputs pedido.
- Todo lo estipulado en el reglamento.

6. Entrega

1. La fecha límite de entrega de la tarea es el día Viernes 12 de Octubre antes de las 23:55 hrs.
2. Para despejar dudas sobre la tarea o el reglamento de tareas puede consultar en la plataforma Moodle en la sección. correspondiente.