Tarea N°1

ICF221 Teoría de la Computación

**Problema 1**

Los servidores de Facebook, Whatsapp e Instagram han sufrido varias caídas este año. Más del 80% de las caídas se han debido a ataques de virus que vienen insertos en archivos multimedia (fotos, videos, audio). Después de un análisis forense de los servidores afectados y de los archivos infectados a nivel binario se determinó que, si los archivos traían ciertas subcadenas binarias y además terminaban en **101**, entonces eran archivos corruptos. Estas subcadenas son **000**0\* o **1111**1\*, es decir, que, si la cadena a analizar tuviese 3 o más ceros consecutivos, o bien, 4 o más unos consecutivos.

Ejemplo 1: (L)impio – (C)orrupto

10100101010100110111000110011100100110011100111 (L)

01001010011101011100100100100110100**1111**10**101** (C)

01001010011101011**000**110010010111011100 (L)

110010011010100100100111011100100110101001001001110**101** (L)

010010100**1111**01011**000**011001001001011100**101** (C)

Obs: el caso extremo de 111101 se considera (C), pues los primeros 4 son la subcadena **1111**1\* y los tres últimos son **101**

El señor Zuckerberg le pide ayuda a los(as) estudiantes de Teoría de la Computación para diseñar un autómata y luego desarrolle un programa en Python que simule el autómata diseñado para que identifique los archivos corruptos y los limpios.

La entrada será un archivo de texto plano que contendrá las secuencias de varios archivos (un archivo por línea) y la salida debe indicar una secuencia de resultados que coincidan con las líneas de archivos.

Ejemplo 2: si el archivo de texto tuviese los 5 archivos (secuencias) del Ejemplo 1 la salida debe ser,

Limpio

Corrupto

Limpio

Limpio

Corrupto

Use el archivo txt con casos de testeo (testP1.txt). No serán los mismo con los que se revisará la tarea

**Problema 2**

La Planificación Automática (PA) es un área de la Inteligencia Artificial en donde un agente autónomo planifica una serie de acciones futuras para lograr un objetivo a partir de una situación inicial. Un plan consiste en un conjunto de acciones, por ejemplo, un robot que recoge objetos de una habitación y los lleva a otra puede realizar las siguientes acciones:

T: Tomar

S: Soltar

A: Avanzar

D: Girar a la derecha

I: Girar a la izquierda

Si suponemos que el robot tiene dos brazos, es decir puede tomar **hasta** 2 objetos **de manera consecutiva** en una habitación para luego soltarlos **en otra**, estos son ejemplos de planes válidos:

* AAAAATDDAAAAATIIAAAAAAAASDAAAS
* AADDATDDAAAAASIIAAAAAAAADAAAAA
* AADDAADDAAAAAAIIAAAAAAAADAAAAA

En los planes válidos si el robot toma 2 objetos, tiene que soltar los 2 objetos. Si el robot toma un objeto, tiene que soltar 1 objeto. Entre las acciones de tomar y soltar debe al menos una acción de avanzar. Estos son ejemplo de planes no validos:

* AAAAATDDAAAAASIIAAAAAAAATDAAASAA
* AADDATDDAAAAASSIIAAAAAAAADAAAAAA
* AADDAASDAAAAAAIIAAATAAAADAAAAADD
* AADDAATDAAAAAAIIAAATAAAADAAAAADD

Cree un autómata finito determinista que permita reconocer planes válidos. El alfabeto del autómata está compuesto las acciones {T, S, A, D, I}. Desarrolle un programa en Python que simule el autómata diseñado para que identifique los planes válidos y los no válidos.

La entrada será un archivo de texto plano que contendrá las secuencias de varios planes (un plan por línea) y la salida debe indicar una secuencia de resultados que coincidan con las líneas de archivos.