School of Computer Science
Faculty of Science
National University of Engineering

Test 1

Subject: Computational Mathematics **Period: 2021-1**

---

1. (4 pts.) Tu tarea es escribir un simple programa que simule jugar *michi* con el usuario. Para hacerlo más fácil, Hemos decidido simplificar el juego. Aquí están nuestras reglas:

   - La maquina (por ejemplo, el programa) jugará utilizando las `X`'s.
   - El usuario jugará utilizando las `O`'s.
   - El primer movimiento es de la maquina: siempre coloca una `X` en el centro del tablero.
   - Todos los cuadros están numerados comenzando con el 1 (observa el ejemplo para que tengas una referencia).
   - El usuario ingresa su movimiento introduciendo el numero de cuadro elegido. El numero debe de ser valido, por ejemplo un valor entero mayor que 0 y menor que 10, y no puede ser un cuadro que ya esté ocupado.
   - El programa verifica si el juego ha terminado. Existen cuatro posibles veredictos: el juego continua, el juego termina en empate, tu ganas, o la maquina gana.
   - La maquina responde con su movimiento y se verifica el estado del juego.

- No se debe implementar algún tipo de inteligencia artificial, la maquina elegirá un cuadro de manera aleatoria, eso es suficiente para este juego.

Implementa las siguientes características:

- El tablero debe ser almacenado como una lista de tres elementos, mientras que cada elemento es otra lista de tres elementos (la lista interna representa las filas) de manera que todos los cuadros puedas ser accedidos empleado la siguiente sintaxis: `board[fila][columna]`
- Cada uno de los elementos internos de la lista puede contener `O`, `X`, o un digito representando el número del cuadro (dicho cuadro se considera como libre).
- Implementa las siguientes funciones:

```
def DisplayBoard(board):
#
# la función acepta un parámetro el cual contiene el estado actual del t
# y lo muestra en la consola
#

def EnterMove(board):
#
# la función acepta el estado actual del tablero y pregunta al usuario a
# verifica la entrada y actualiza el tablero acorde a la decisión del us
#

def MakeListOfFreeFields(board):
#
# la función examina el tablero y construye una lista de todos los cuadr
# la lista esta compuesta por tuplas, cada tupla es un par de números qu
#
```

```
def VictoryFor(board, sign):
#
# la función analiza el estatus del tablero para verificar si
# el jugador que utiliza las Os o las Xs ha ganado el juego
#

def DrawMove(board):
#
# la función dibuja el movimiento de la maquina y actualiza el tablero
#
```

Para obtener un valor numérico aleatorio se puede emplear una función integrada de Python denominada `randrange()`. El siguiente ejemplo muestra como utilizarla (El programa imprime 10 números aleatorios del 1 al 8):

```
from random import randrange

for i in range(10):
   print(randrange(8))
```

2. (4 pts.) Seguramente has visto un visualizador de siete segmentos. Es un dispositivo (a veces electrónico, a veces mecánico) diseñado para presentar un dígito decimal utilizando un subconjunto de siete segmentos.
   Escribe un programa que puede simular el funcionamiento de un visualizador de siete segmentos, aunque vas a usar LEDs individuales en lugar de segmentos. Cada dígito es construido con 13 LEDs (algunos iluminados, otros apagados, por supuesto), así es como lo imaginamos:

```
# ### ### # # ### ### ### ### ### ###
#   #   # # # #   #     # # # # # # #
# ### ### ### ### ###   # ### ### # #
```

```
# #     #   #   # # #   # # #   # # #
# ### ###   # ### ###   # ### ### ###
```

N.B.: el número 8 muestra todas las luces LED encendidas.
Tu código debe mostrar cualquier número entero no negativo ingresado por el usuario.
Sugerencia: puede ser muy útil usar una lista que contenga patrones de los diez dígitos.

3. (4 pts.) The three points $P_0 = (1, 4, 8)$, $P_1 = (2, 2, -1)$, and $P_2 = (5, 4, 0)$ are three lines determining a plane. Using homogeneous coordinates, find the vector describing the plane. In terms of Cartesian coordinates, give the normal to the plane. Now, consider three planes with implicit equations $2 \cdot x + 3 \cdot y - 6 \cdot z = 2$, $-x + 4 \cdot y + z = 7$, and $-8 \cdot x + y - z = 10$. Using homogeneous coordinates, find the point of intersection. Convert the result to Cartesian coordinates.

4. (5 pts.) Write a method (function) that takes the eight vertices of a box (not necessarily a parallelepiped) plus a point as input and outputs whether the point is inside the box. You must include a code to determine whether the box is well formed.

May 5, 2021