UNIVERSITY OF JOHANNESBURG

FACULTY OF SCIENCE

COMPUTER SCIENCE 1A **SAMPLE DESIGN**

Problem Description

Design a program that will output a well-known Cellular Automata, rule 30. This version of the cellular automaton uses an array of elements that are considered to be either alive or dead.
Ref: Practicaal Assessment 7

Input & Output

Input	
Input Description	*Mechanism*
New value of array	Standard Input Stream
Number of rounds	Standard Input Stream
Output	
Output Description	*Stream (optional)*
New sequence	Standard output stream

Data Format

Identifier	*Data Type*	*Description*
intPrevInd	Integer	Tracks where the previous value in the array is
intNextInd	Integer	Tracks where the next value in the array is
intPrevNum	Integer	Tracks the previous value in the array
intNextNum	Integer	Tracks the next value in the array
arrNums	Array of Integers	Stores the sequence

Pseudo Code

```
For every round:
   Run function CompltAutomation
   Run function PrintArray


CompltAutomation:
      For all the values in the array:
         integar intPrevInd → n-1
         integar intNextInd → n+1
```

```
        if n is less than or equalto 0 then intPrevInd = intCount-1

        if n is the last value in the index of the array then intNextInd = 0

        integar intPrevNum → arrNums[intPrevInd]
        integar intNextNum → arrNums[intNextInd]

        // Part 1
        If intPrevNum is equal to ALIVE and intNextNum is equal to ALIVE then:
           If arrNums[n] is equal to ALIVE then:
              arrNums[n] → DEAD

        // Part 2
        If intPrevNum is equal to ALIVE and intNextNum is equal to DEAD then:
           If arrNums[n] is equal to ALIVE then:
              arrNums[n] → DEAD

        // Part 3
        If intPrevNum is equal to ALIVE and intNextNum is equal to ALIVE then:
           If arrNums[n] is equal to DEAD then:
              arrNums[n] → DEAD

        // Part 4
        If intPrevNum is equal to ALIVE and intNextNum is equal to DEAD then:
             If arrNums[n] is equal to DEAD then:
              arrNums[n] → ALIVE

        // Part 5
        If intPrevNum is equal to DEAD and intNextNum is equal to ALIVE then:
           If arrNums[n] is equal to ALIVE then:
              arrNums[n] → ALIVE

        // Part 6
        If intPrevNum is equal to DEAD and intNextNum is equal to DEAD then:
           If arrNums[n] is equal to ALIVE then:
              arrNums[n] → ALIVE

        // Part 7
        If intPrevNum is equal to DEAD and intNextNum is equal to ALIVE then:
           If arrNums[n] is equal to DEAD then:
              arrNums[n] → ALIVE

        // Part 8
        If intPrevNum is equal to DEAD and intNextNum is equal to DEAD then:
           If arrNums[n] is equal to DEAD then:
              arrNums[n] → DEAD


PrintArray:
      For all the values in the array:
         If arrNums[n] is equal to DEAD then display charDEAD

          If arrNums[n] is equal to ALIVE then display charALIVE
```

UML Activity Diagram