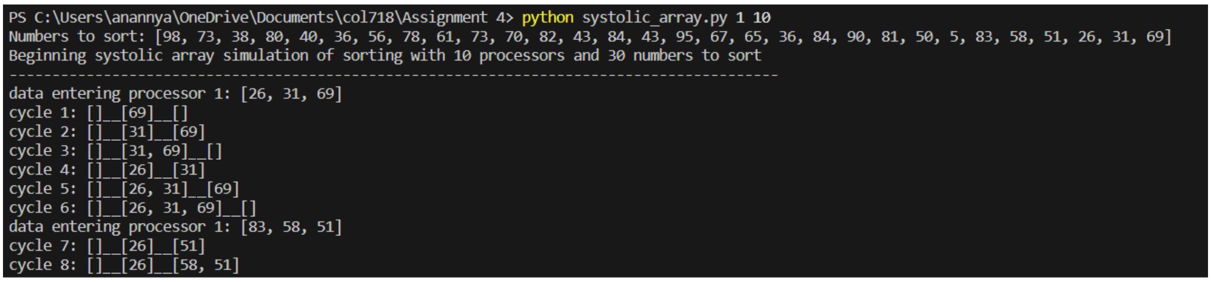
Systolic Array Simulation

Anannya Mathur

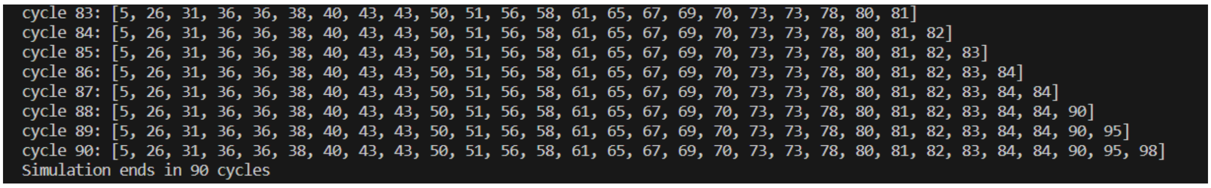
Parameter m=The size of the systolic array.
m, which denotes the number of processors available, is taken as the input, while the size of the arrays, N, is fixed

1. Sorting:

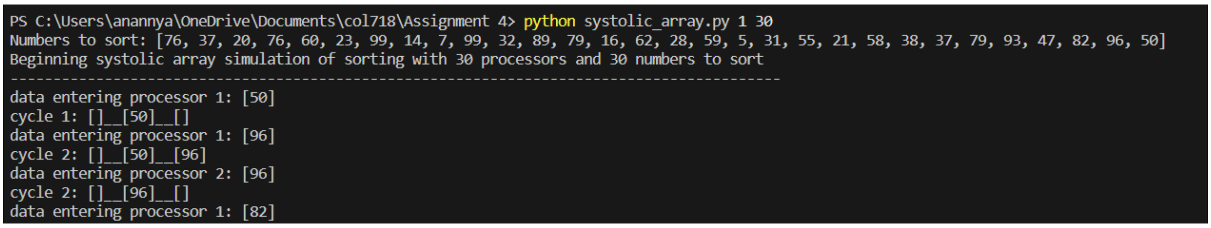
   N is fixed at 30.



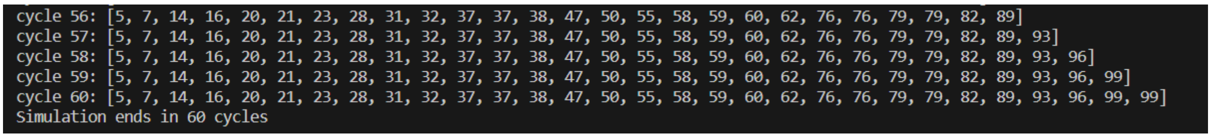
```
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> python systolic_array.py 1 10
Numbers to sort: [98, 73, 38, 80, 40, 36, 56, 78, 61, 73, 70, 82, 43, 84, 43, 95, 67, 65, 36, 84, 90, 81, 50, 5, 83, 58, 51, 26, 31, 69]
Beginning systolic array simulation of sorting with 10 processors and 30 numbers to sort
-----------------------------------------------------------------------------------------
data entering processor 1: [26, 31, 69]
cycle 1: []__[69]__[]
cycle 2: []__[31]__[69]
cycle 3: []__[31, 69]__[]
cycle 4: []__[26]__[31]
cycle 5: []__[26, 31]__[69]
cycle 6: []__[26, 31, 69]__[]
data entering processor 1: [83, 58, 51]
cycle 7: []__[26]__[51]
cycle 8: []__[26]__[58, 51]
```



```
  cycle 83: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81]
  cycle 84: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82]
  cycle 85: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83]
  cycle 86: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83, 84]
  cycle 87: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83, 84, 84]
  cycle 88: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83, 84, 84, 90]
  cycle 89: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83, 84, 84, 90, 95]
  cycle 90: [5, 26, 31, 36, 36, 38, 40, 43, 43, 50, 51, 56, 58, 61, 65, 67, 69, 70, 73, 73, 78, 80, 81, 82, 83, 84, 84, 90, 95, 98]
  Simulation ends in 90 cycles
```

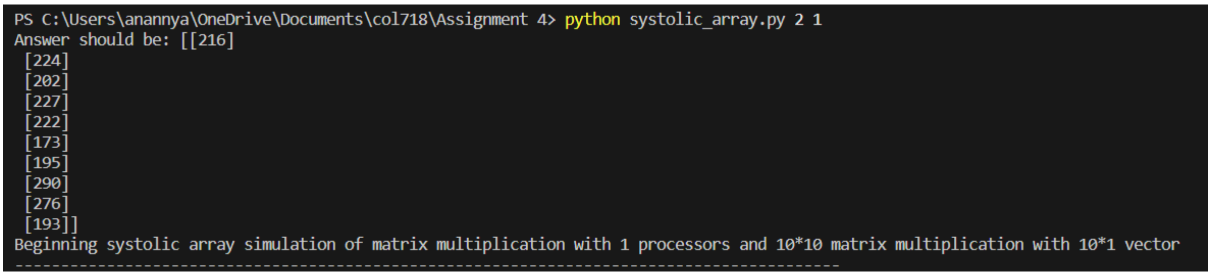


```
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> python systolic_array.py 1 30
Numbers to sort: [76, 37, 20, 76, 60, 23, 99, 14, 7, 99, 32, 89, 79, 16, 62, 28, 59, 5, 31, 55, 21, 58, 38, 37, 79, 93, 47, 82, 96, 50]
Beginning systolic array simulation of sorting with 30 processors and 30 numbers to sort
-----------------------------------------------------------------------------------------
data entering processor 1: [50]
cycle 1: []__[50]__[]
data entering processor 1: [96]
cycle 2: []__[50]__[96]
data entering processor 2: [96]
cycle 2: []__[96]__[]
data entering processor 1: [82]
```

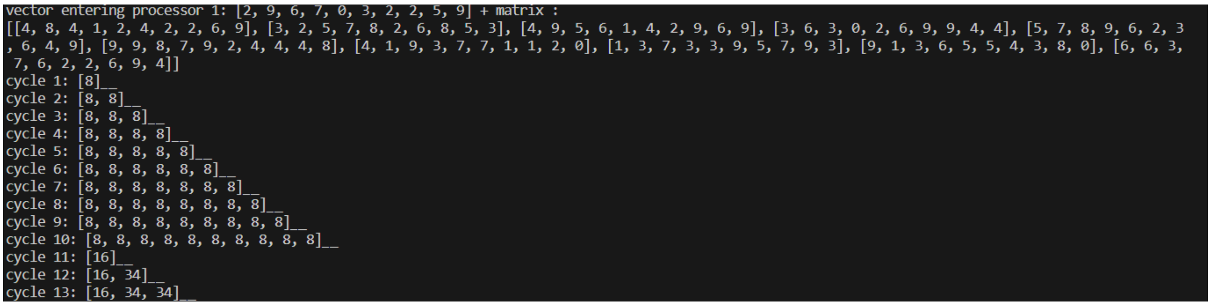


```
 cycle 56: [5, 7, 14, 16, 20, 21, 23, 28, 31, 32, 37, 37, 38, 47, 50, 55, 58, 59, 60, 62, 76, 76, 79, 79, 82, 89]
 cycle 57: [5, 7, 14, 16, 20, 21, 23, 28, 31, 32, 37, 37, 38, 47, 50, 55, 58, 59, 60, 62, 76, 76, 79, 79, 82, 89, 93]
 cycle 58: [5, 7, 14, 16, 20, 21, 23, 28, 31, 32, 37, 37, 38, 47, 50, 55, 58, 59, 60, 62, 76, 76, 79, 79, 82, 89, 93, 96]
 cycle 59: [5, 7, 14, 16, 20, 21, 23, 28, 31, 32, 37, 37, 38, 47, 50, 55, 58, 59, 60, 62, 76, 76, 79, 79, 82, 89, 93, 96, 99]
 cycle 60: [5, 7, 14, 16, 20, 21, 23, 28, 31, 32, 37, 37, 38, 47, 50, 55, 58, 59, 60, 62, 76, 76, 79, 79, 82, 89, 93, 96, 99, 99]
 Simulation ends in 60 cycles
```

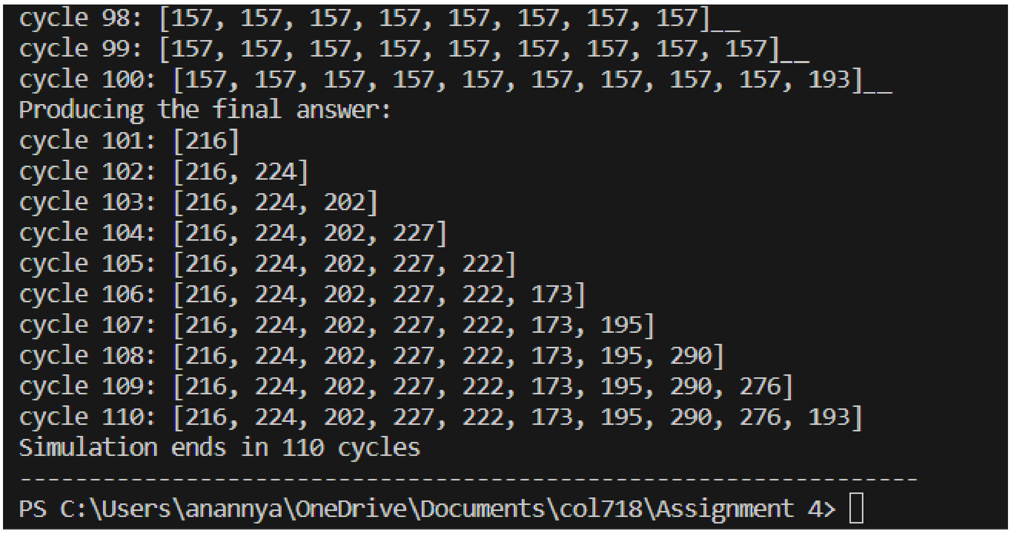
2. Matrix-Vector multiplication:



```
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> python systolic_array.py 2 1
Answer should be: [[216]
 [224]
 [202]
 [227]
 [222]
 [173]
 [195]
 [290]
 [276]
 [193]]
Beginning systolic array simulation of matrix multiplication with 1 processors and 10*10 matrix multiplication with 10*1 vector
-----------------------------------------------------------------------------------------
```



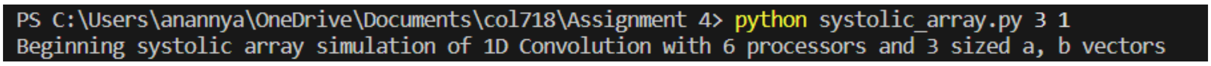
```
vector entering processor 1: [2, 9, 6, 7, 0, 3, 2, 2, 5, 9] + matrix :
[[4, 8, 4, 1, 2, 4, 2, 2, 6, 9], [3, 2, 5, 7, 8, 2, 6, 8, 5, 3], [4, 9, 5, 6, 1, 4, 2, 9, 6, 9], [3, 6, 3, 0, 2, 6, 9, 9, 4, 4], [5, 7, 8, 9, 6, 2, 3
, 6, 4, 9], [9, 9, 8, 7, 9, 2, 4, 4, 4, 8], [4, 1, 9, 3, 7, 7, 1, 1, 2, 0], [1, 3, 7, 3, 3, 9, 5, 7, 9, 3], [9, 1, 3, 6, 5, 5, 4, 3, 8, 0], [6, 6, 3,
 7, 6, 2, 2, 6, 9, 4]]
cycle 1: [8]__
cycle 2: [8, 8]__
cycle 3: [8, 8, 8]__
cycle 4: [8, 8, 8, 8]__
cycle 5: [8, 8, 8, 8, 8]__
cycle 6: [8, 8, 8, 8, 8, 8]__
cycle 7: [8, 8, 8, 8, 8, 8, 8]__
cycle 8: [8, 8, 8, 8, 8, 8, 8, 8]__
cycle 9: [8, 8, 8, 8, 8, 8, 8, 8, 8]__
cycle 10: [8, 8, 8, 8, 8, 8, 8, 8, 8, 8]__
cycle 11: [16]__
cycle 12: [16, 34]__
cycle 13: [16, 34, 34]__
```



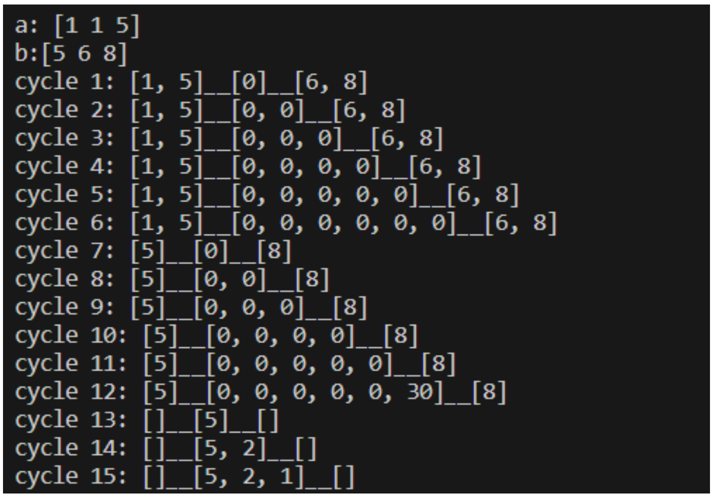
```
cycle 98: [157, 157, 157, 157, 157, 157, 157, 157]__
cycle 99: [157, 157, 157, 157, 157, 157, 157, 157, 157]__
cycle 100: [157, 157, 157, 157, 157, 157, 157, 157, 157, 193]__
Producing the final answer:
cycle 101: [216]
cycle 102: [216, 224]
cycle 103: [216, 224, 202]
cycle 104: [216, 224, 202, 227]
cycle 105: [216, 224, 202, 227, 222]
cycle 106: [216, 224, 202, 227, 222, 173]
cycle 107: [216, 224, 202, 227, 222, 173, 195]
cycle 108: [216, 224, 202, 227, 222, 173, 195, 290]
cycle 109: [216, 224, 202, 227, 222, 173, 195, 290, 276]
cycle 110: [216, 224, 202, 227, 222, 173, 195, 290, 276, 193]
Simulation ends in 110 cycles
-----------------------------------------------------------------
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> 
```

3. 1D Convolution:

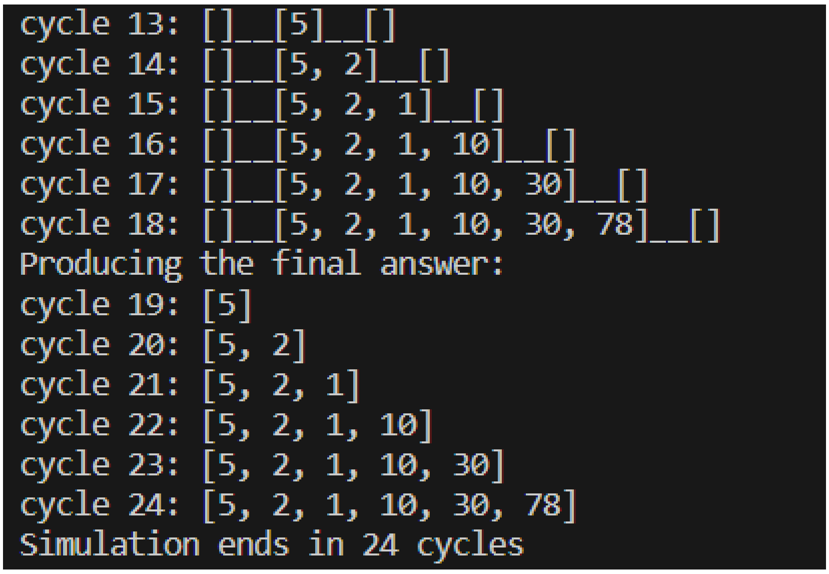
In this case, it is assumed that the number of processors=2*N, where N is the size of the two vectors, a and b.



```
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> python systolic_array.py 3 1
Beginning systolic array simulation of 1D Convolution with 6 processors and 3 sized a, b vectors
```



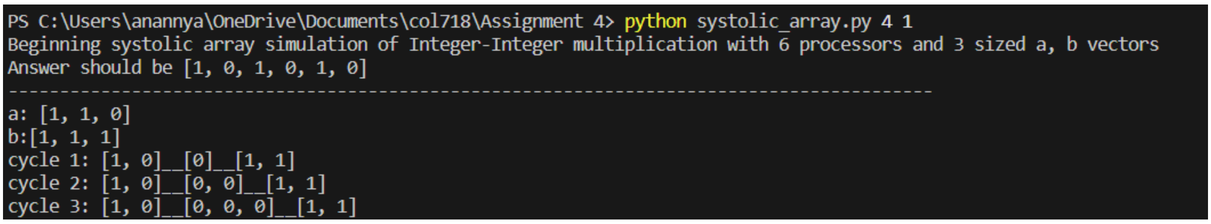
```
a: [1 1 5]
b:[5 6 8]
cycle 1: [1, 5]__[0]__[6, 8]
cycle 2: [1, 5]__[0, 0]__[6, 8]
cycle 3: [1, 5]__[0, 0, 0]__[6, 8]
cycle 4: [1, 5]__[0, 0, 0, 0]__[6, 8]
cycle 5: [1, 5]__[0, 0, 0, 0, 0]__[6, 8]
cycle 6: [1, 5]__[0, 0, 0, 0, 0, 0]__[6, 8]
cycle 7: [5]__[0]__[8]
cycle 8: [5]__[0, 0]__[8]
cycle 9: [5]__[0, 0, 0]__[8]
cycle 10: [5]__[0, 0, 0, 0]__[8]
cycle 11: [5]__[0, 0, 0, 0, 0]__[8]
cycle 12: [5]__[0, 0, 0, 0, 0, 30]__[8]
cycle 13: []__[5]__[]
cycle 14: []__[5, 2]__[]
cycle 15: []__[5, 2, 1]__[]
```



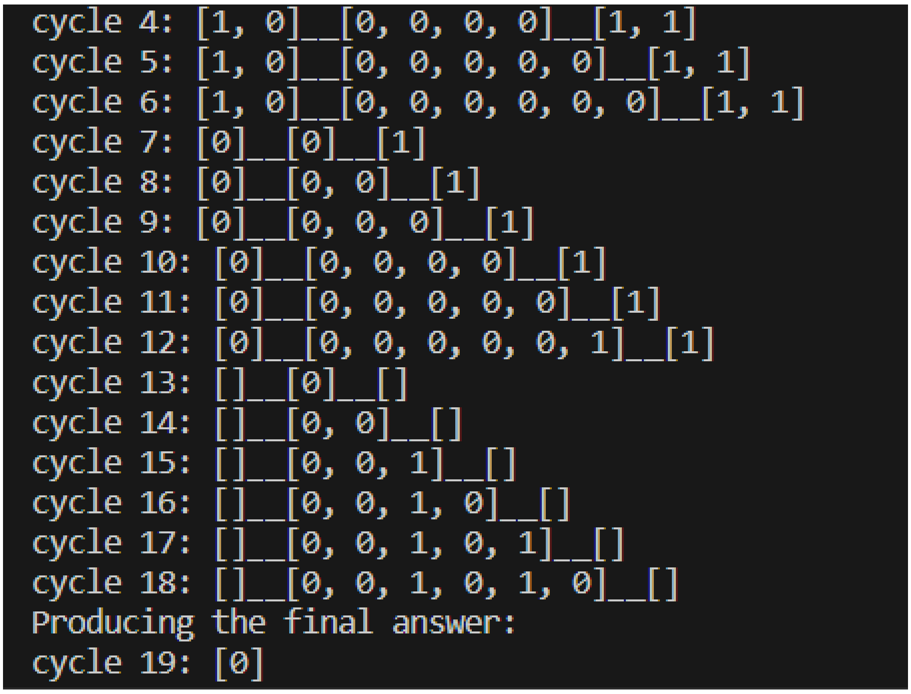
```
cycle 13: []__[5]__[]
cycle 14: []__[5, 2]__[]
cycle 15: []__[5, 2, 1]__[]
cycle 16: []__[5, 2, 1, 10]__[]
cycle 17: []__[5, 2, 1, 10, 30]__[]
cycle 18: []__[5, 2, 1, 10, 30, 78]__[]
Producing the final answer:
cycle 19: [5]
cycle 20: [5, 2]
cycle 21: [5, 2, 1]
cycle 22: [5, 2, 1, 10]
cycle 23: [5, 2, 1, 10, 30]
cycle 24: [5, 2, 1, 10, 30, 78]
Simulation ends in 24 cycles
```

4. Integer-Integer Multiplication:

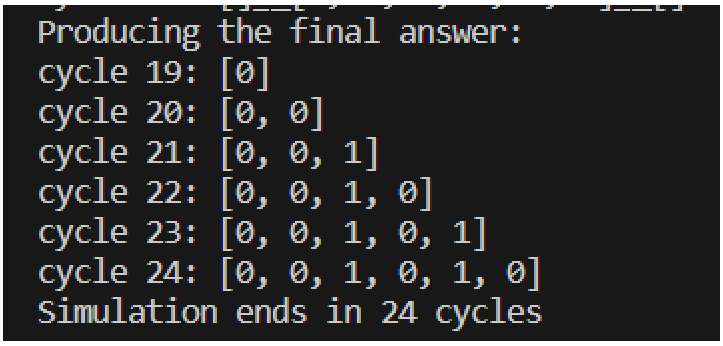
In this case, it is assumed that the number of processors=2*N, where N is the size of the two vectors, a and b.



```
PS C:\Users\anannya\OneDrive\Documents\col718\Assignment 4> python systolic_array.py 4 1
Beginning systolic array simulation of Integer-Integer multiplication with 6 processors and 3 sized a, b vectors
Answer should be [1, 0, 1, 0, 1, 0]
------------------------------------------------------------------------------------------
a: [1, 1, 0]
b:[1, 1, 1]
cycle 1: [1, 0]__[0]__[1, 1]
cycle 2: [1, 0]__[0, 0]__[1, 1]
cycle 3: [1, 0]__[0, 0, 0]__[1, 1]
```



```
cycle 4: [1, 0]__[0, 0, 0, 0]__[1, 1]
cycle 5: [1, 0]__[0, 0, 0, 0, 0]__[1, 1]
cycle 6: [1, 0]__[0, 0, 0, 0, 0, 0]__[1, 1]
cycle 7: [0]__[0]__[1]
cycle 8: [0]__[0, 0]__[1]
cycle 9: [0]__[0, 0, 0]__[1]
cycle 10: [0]__[0, 0, 0, 0]__[1]
cycle 11: [0]__[0, 0, 0, 0, 0]__[1]
cycle 12: [0]__[0, 0, 0, 0, 0, 1]__[1]
cycle 13: []__[0]__[]
cycle 14: []__[0, 0]__[]
cycle 15: []__[0, 0, 1]__[]
cycle 16: []__[0, 0, 1, 0]__[]
cycle 17: []__[0, 0, 1, 0, 1]__[]
cycle 18: []__[0, 0, 1, 0, 1, 0]__[]
Producing the final answer:
cycle 19: [0]
```



```
Producing the final answer:
cycle 19: [0]
cycle 20: [0, 0]
cycle 21: [0, 0, 1]
cycle 22: [0, 0, 1, 0]
cycle 23: [0, 0, 1, 0, 1]
cycle 24: [0, 0, 1, 0, 1, 0]
Simulation ends in 24 cycles
```