Question 1: Using the hash function ‘key mod 7’, insert the following sequence of keys in the hash table-

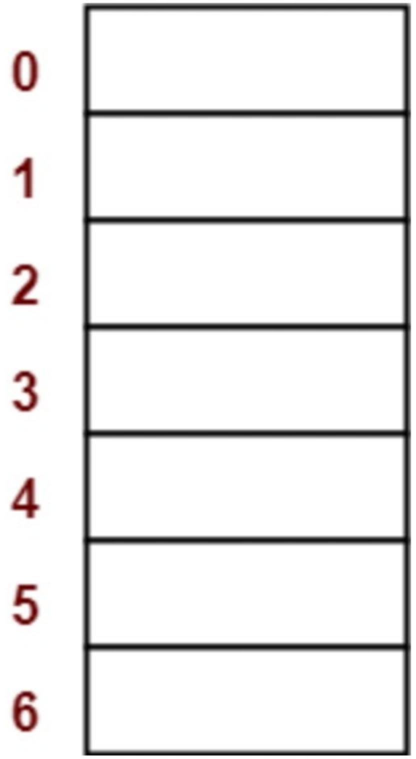
50, 700, 76, 85, 92, 73 and 101

Use chaining technique for collision resolution.

Solution

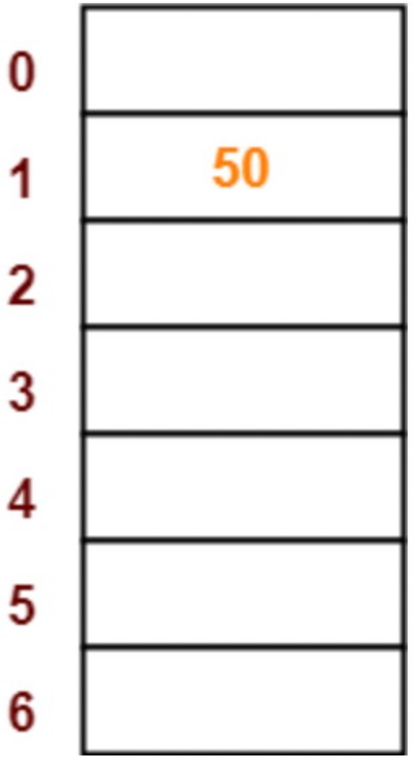
Step-01:

- Draw an empty hash table.
- For the given hash function, the possible range of hash values is [0, 6].
- So, draw an empty hash table consisting of 7 buckets as-



Step-02:

- Insert the given keys in the hash table one by one.
- The first key to be inserted in the hash table = 50.
- Bucket of the hash table to which key 50 maps = 50 mod 7 = 1.
- So, key 50 will be inserted in bucket-1 of the hash table as-



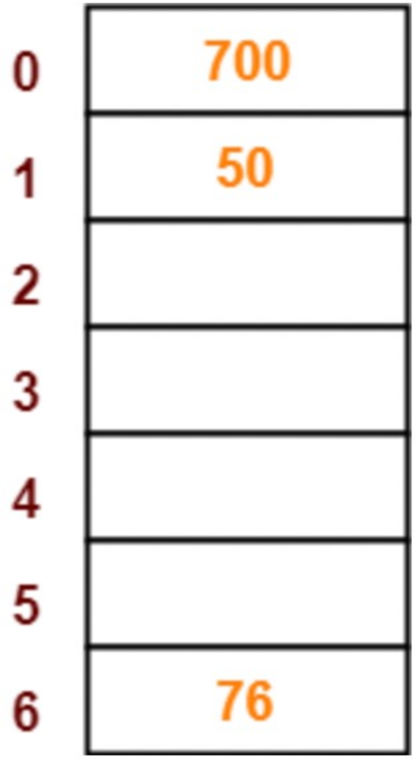
Step-03:

- The next key to be inserted in the hash table = 700.
- Bucket of the hash table to which key 700 maps = 700 mod 7 = 0.
- So, key 700 will be inserted in bucket-0 of the hash table as-

0	700
1	50
2	
3	
4	
5	
6	

Step-04:

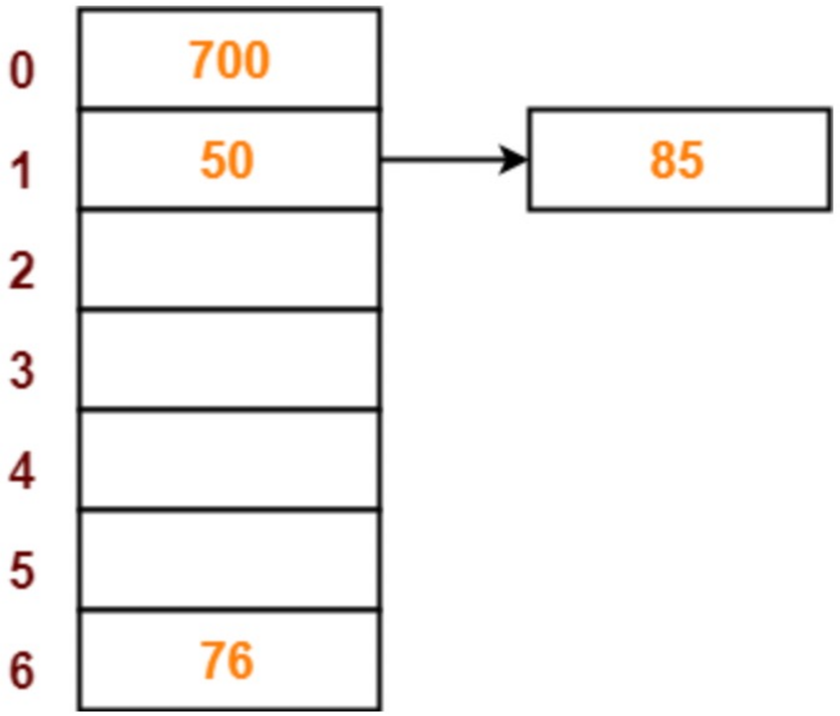
- The next key to be inserted in the hash table = 76.
- Bucket of the hash table to which key 76 maps = 76 mod 7 = 6.
- So, key 76 will be inserted in bucket-6 of the hash table as-



0	700
1	50
2	
3	
4	
5	
6	76

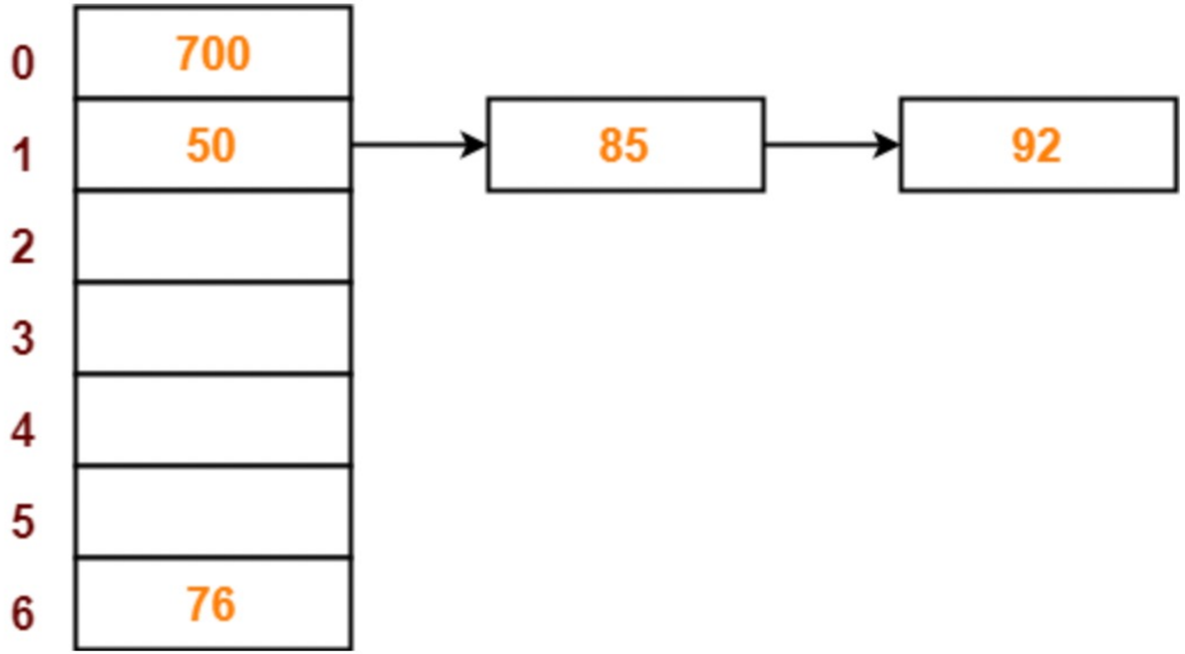
Step-05:

- The next key to be inserted in the hash table = 85.
- Bucket of the hash table to which key 85 maps = 85 mod 7 = 1.
- Since bucket-1 is already occupied, so collision occurs.
- Separate chaining handles the collision by creating a linked list to bucket-1.
- So, key 85 will be inserted in bucket-1 of the hash table as-



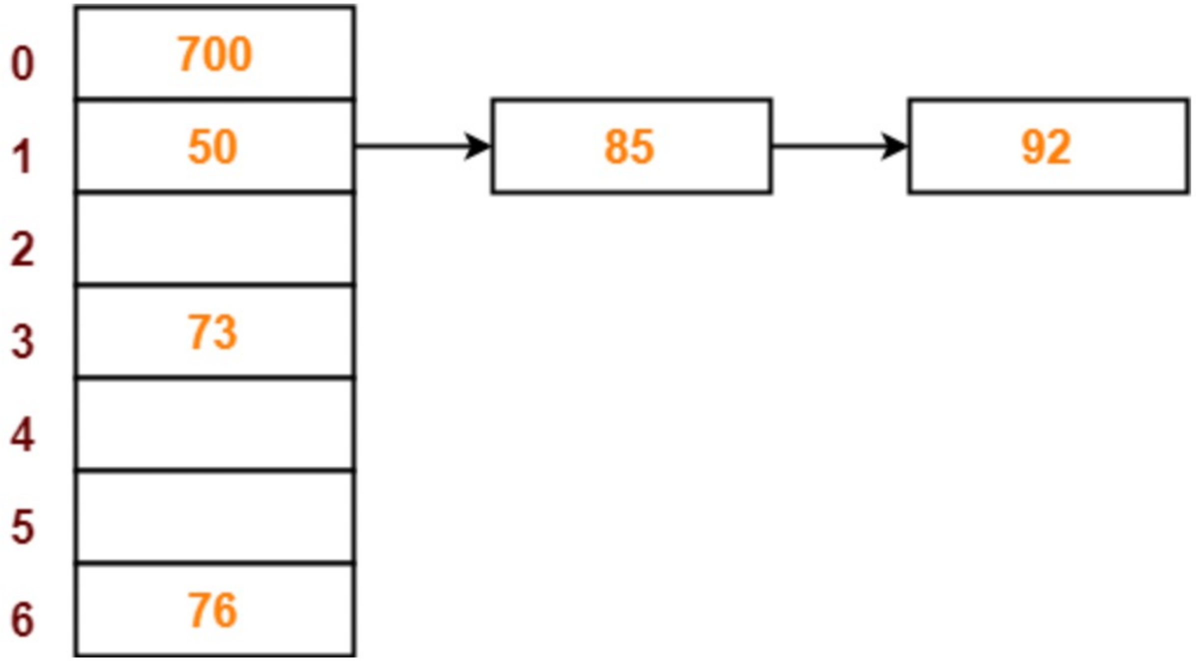
Step-06:

- The next key to be inserted in the hash table = 92.
- Bucket of the hash table to which key 92 maps = 92 mod 7 = 1.
- Since bucket-1 is already occupied, so collision occurs.
- Separate chaining handles the collision by creating a linked list to bucket-1.
- So, key 92 will be inserted in bucket-1 of the hash table as-



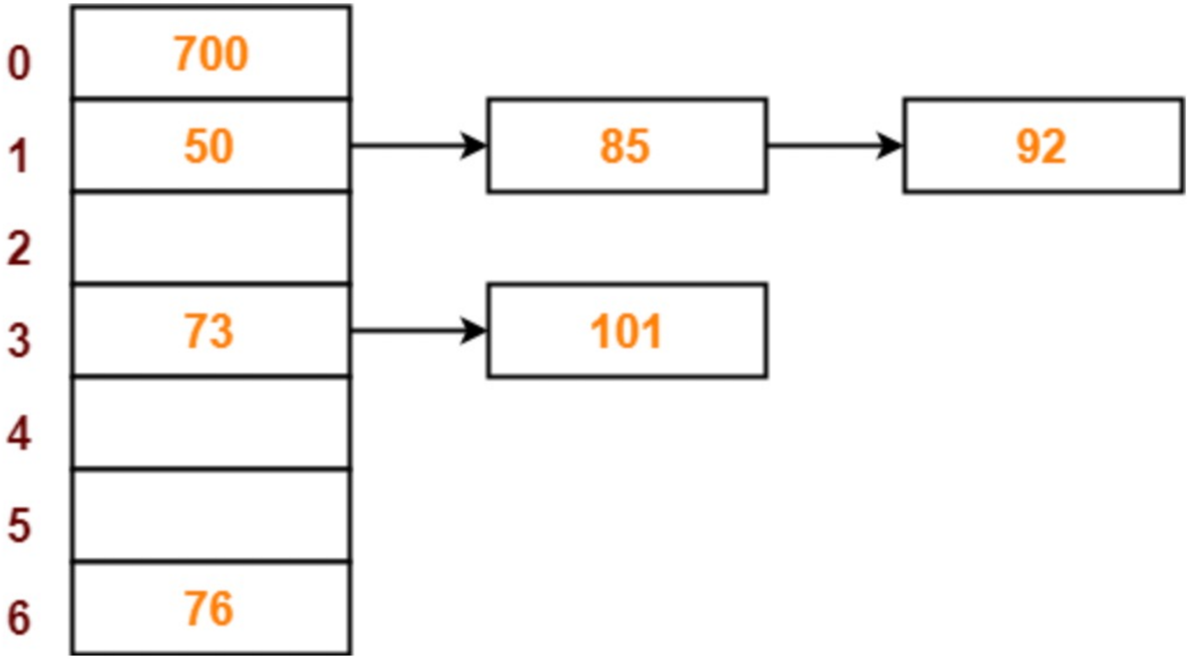
Step-07:

- The next key to be inserted in the hash table = 73.
- Bucket of the hash table to which key 73 maps = 73 mod 7 = 3.
- So, key 73 will be inserted in bucket-3 of the hash table as-



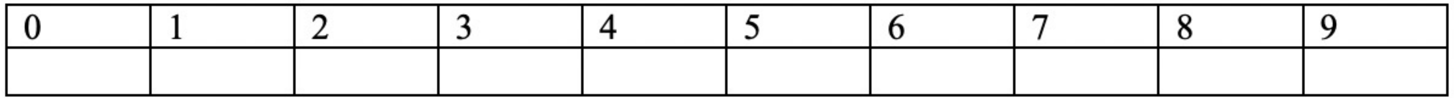
Step-08:

- The next key to be inserted in the hash table = 101.
- Bucket of the hash table to which key 101 maps = 101 mod 7 = 3.
- Since bucket-3 is already occupied, so collision occurs.
- Separate chaining handles the collision by creating a linked list to bucket-3.
- So, key 101 will be inserted in bucket-3 of the hash table as-



Try yourself!!!

Question 2: Insert the following values into a hash table of size 10 using the hash equation $(x^2 +1)$ % 10 using the separate chaining technique. Insert these values in sequential order: 1,2,5,6, 8.

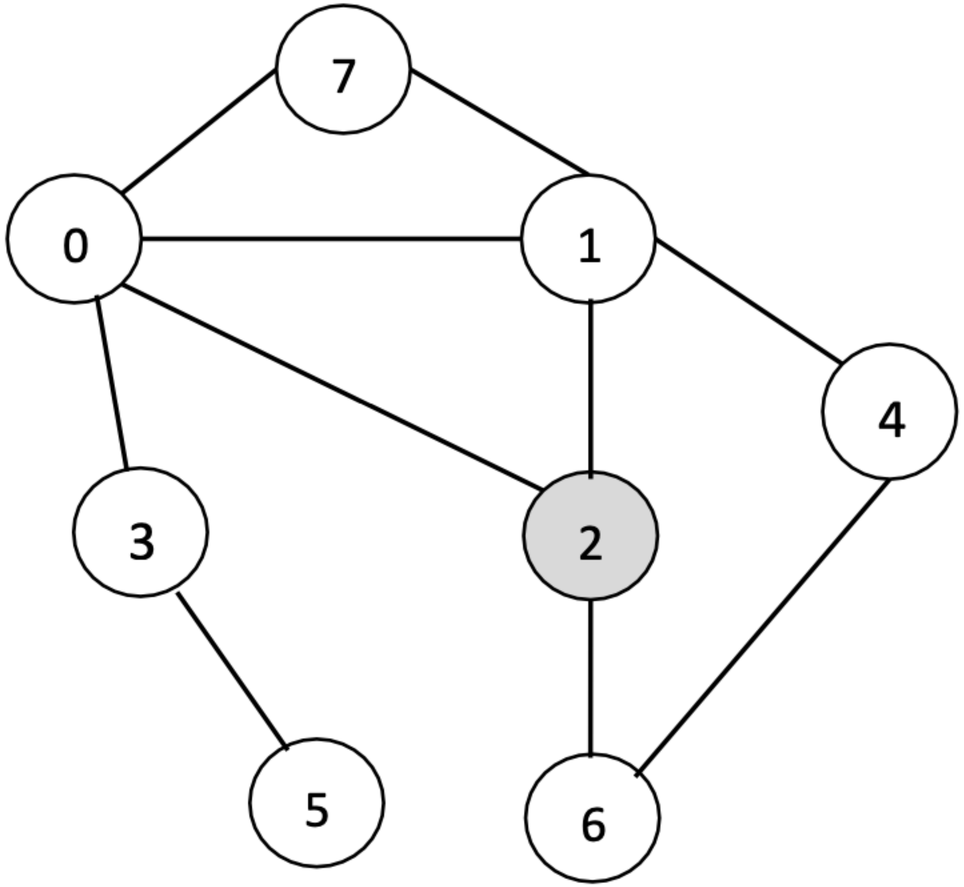


0	1	2	3	4	5	6	7	8	9

Try yourself!!!

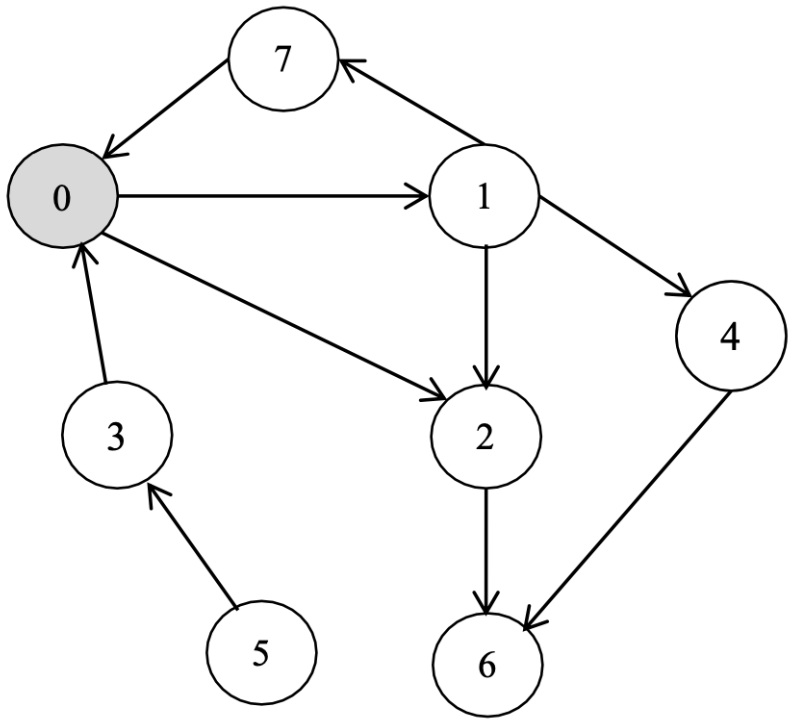
Question 3: Given the input {4371, 1323, 6173, 4199, 4344, 9679, 1989}, a fixed table size of 10, and a hash function H(X) = X mod 10, show the resulting chaining hash table

Question: Traverse the graph below in Breadth-First order, starting from vertex 2. Whenever you examine neighbors of a vertex, process them in increasing order of vertex number. Half the points will be lost if you visit the neighbors in a different order.



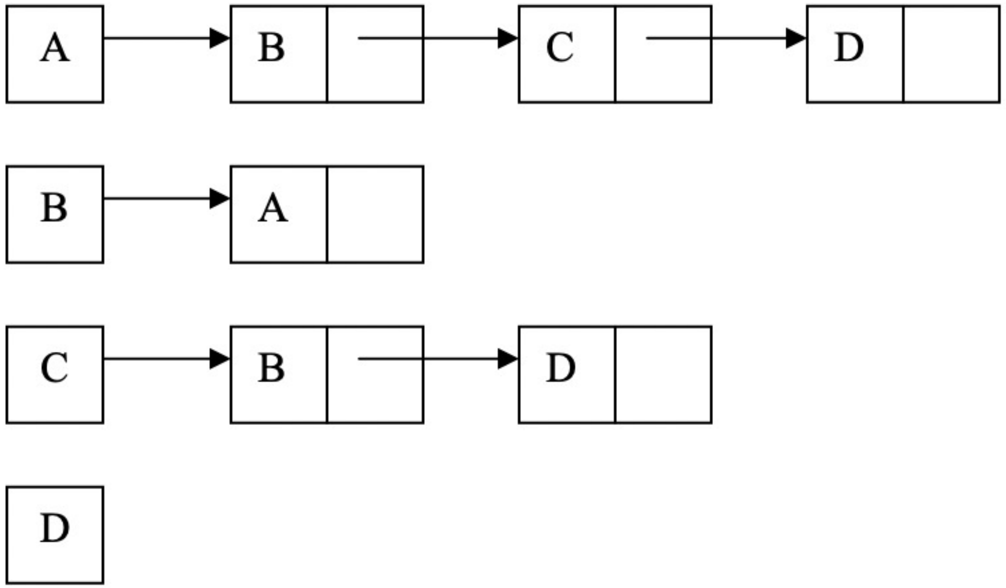
Visited Vertex: 2, 0, 1, 6, 3, 7, 4, 5

Question: Traverse this graph with Depth First Search (DFS) (all DFS, not just one DFS-Visit). If a vertex has more than one neighbors you must visit them in increasing order of vertex number, for example from node 1 you would visit the neighbors in order: 2,4,7. Half the points will be lost if you visit the neighbors in a different order.



Visited vertex
0
1
2
6
4
7
3
5

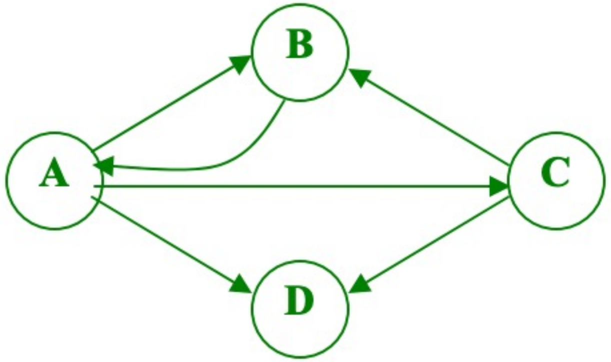
Question: Here is an adjacency list representation of a directed graph where there are no weights assigned to the edges).



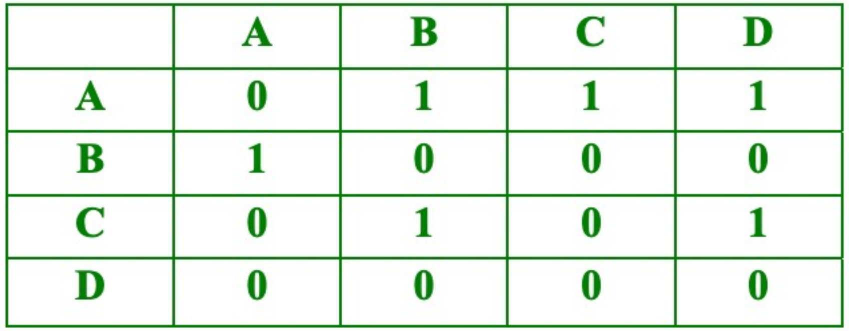
(a) Draw a picture of the directed graph that has the above adjacency list representation.

(b) Another way to represent a graph is an adjacency matrix. Draw the adjacency matrix for this graph.

(a) Draw a picture of the directed graph that has the above adjacency list representation.



(b) Another way to represent a graph is an adjacency matrix. Draw the adjacency matrix for this graph.



	A	B	C	D
A	0	1	1	1
B	1	0	0	0
C	0	1	0	1
D	0	0	0	0