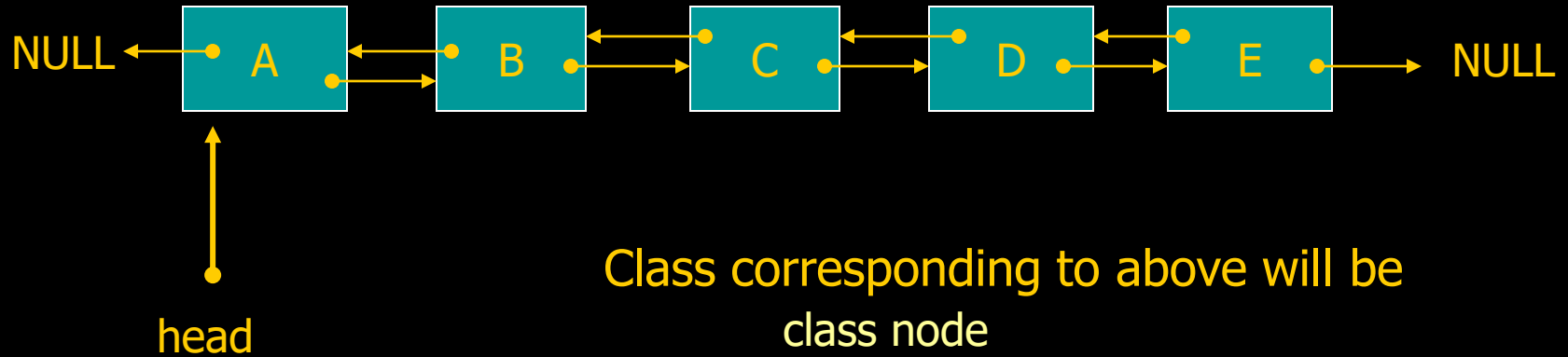


Lecture # 4

Doubly Link List

- In a **Doubly linked list**, each **item** is allocated space as it is added to the list. A **link** is kept with each item to the **next item** and **previous item** in the list.
- Each node of the list has **three** elements:
 - The **item** being stored in the list
 - A pointer to the **next** item in the list
 - A pointer to the **previous** item in the list
- The last node in the list contains a **NULL** pointer (**next**) to indicate that it is the **end** or **tail** of the list and first node contains **NULL** pointer for **previous** pointer to indicate that it is **head** of the Link List.

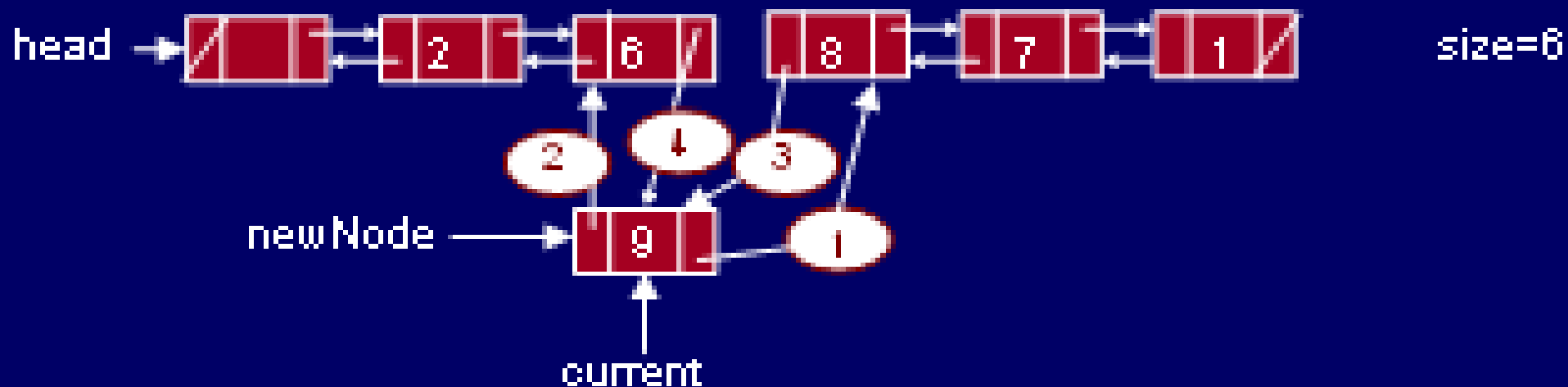
■ Figure showing Doubly Link List



Class corresponding to above will be

```
class node
{
    public:
    char info;
    node *next;
    node *previous;
    node(int val)
    {
        info = val;
        next = NULL;
        previous = NULL;
    }
};
```

Adding element - Doubly-Linked List



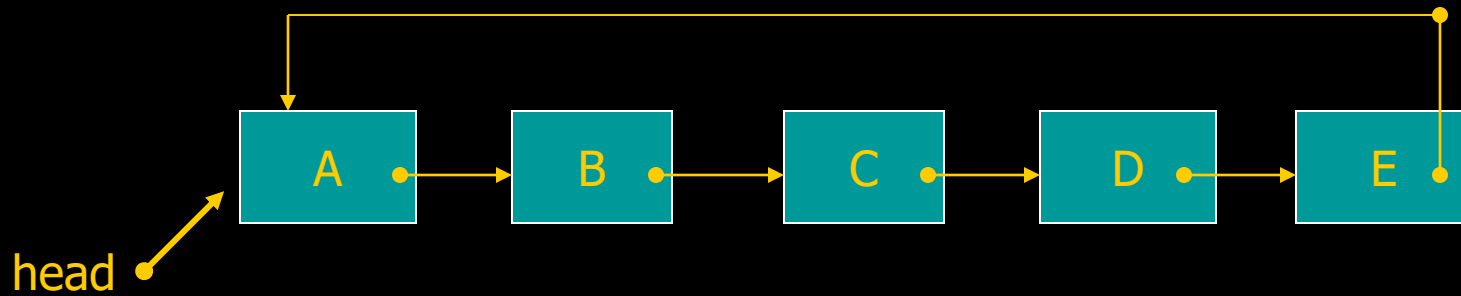
Deletion - Doubly-Linked List

- How
- Think yourself
- Discussion from Implementation (code) point of view.

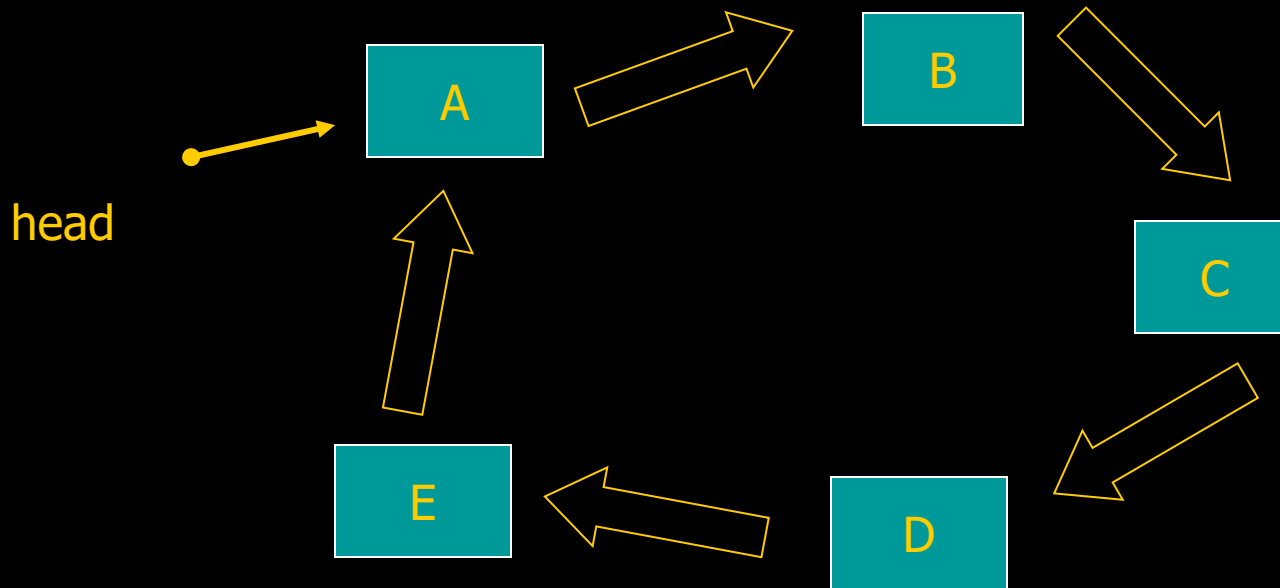
Circular Link List (Ring)

- Ring data structure can be regarded as circular Link List.
- In Link List, instead of making last node's next pointer to NULL, we can assign or point it to head node so that it sounds as ring data structure as follows.

■ Figure showing Ring.....



OR

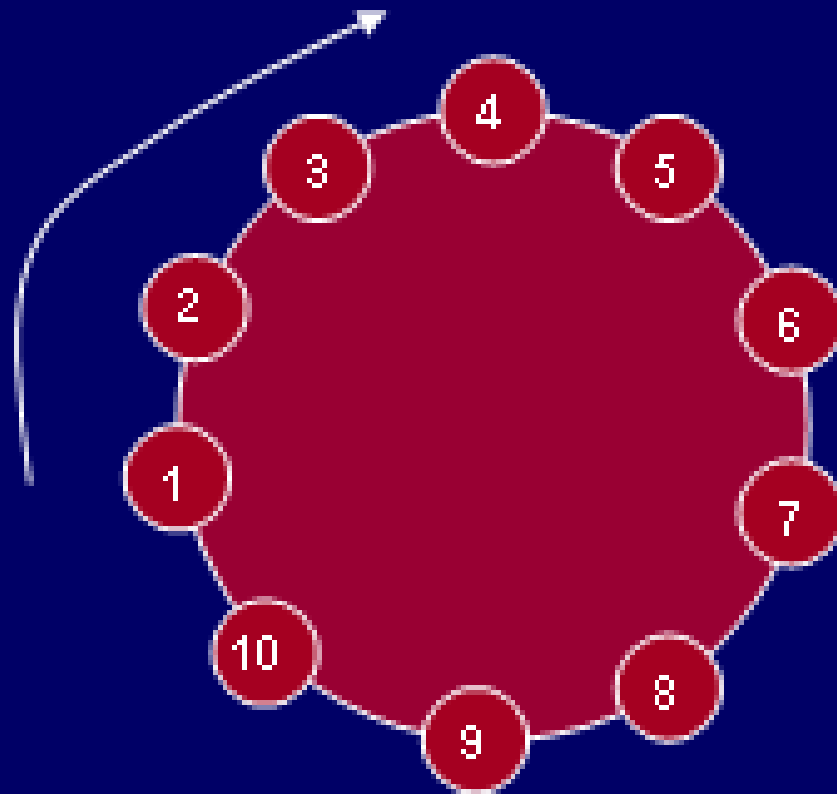


Josephus Problem

- A case where circularly linked list is handy to provide the solution of the *Josephus Problem*.
- Consider there are 10 persons. They would like to choose a leader.
- The way they decide is that all 10 sit in a circle.
- They start a count with person 1 and go in clockwise direction and skip 3. Person 4 reached is eliminated.
- The count starts with the fifth and the next person to go is the fourth in count.
- Eventually, a single person remains.

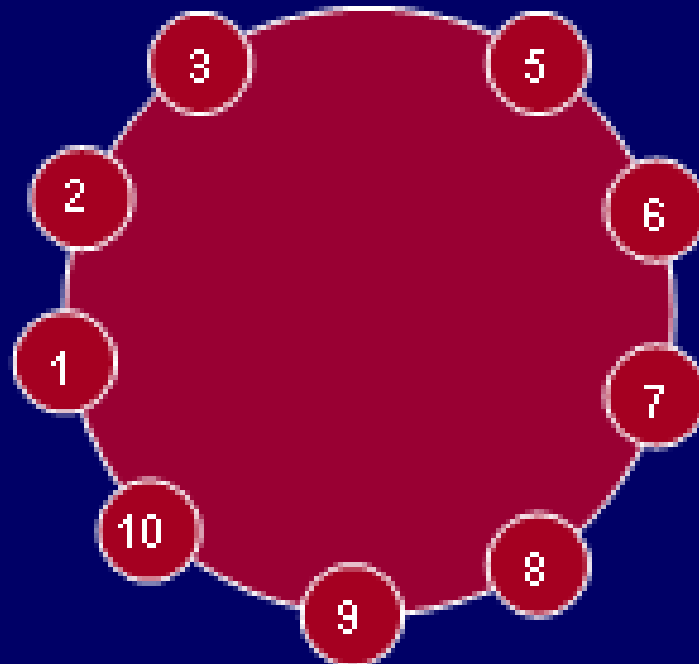
Josephus Problem

- $N=10, M=3$



Josephus Problem

- $N=10, M=3$

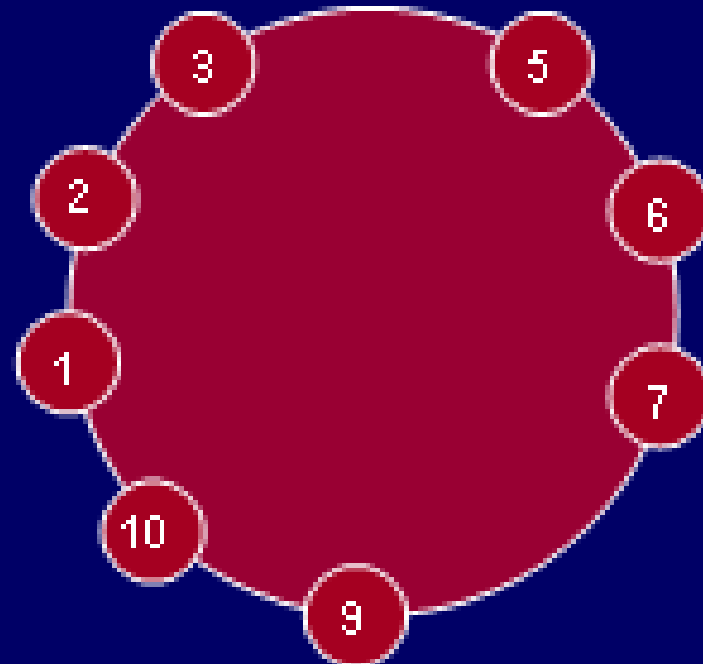


eliminated



Josephus Problem

- $N=10, M=3$

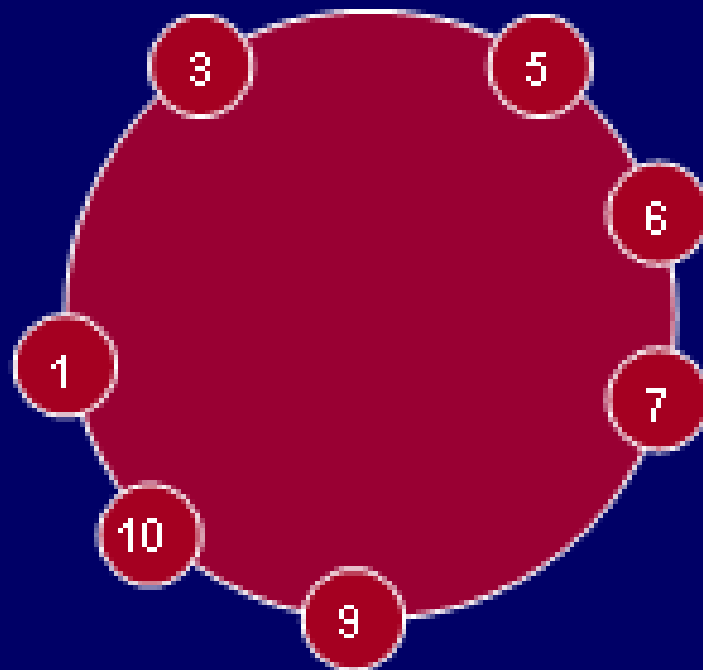


eliminated



Josephus Problem

- $N=10, M=3$

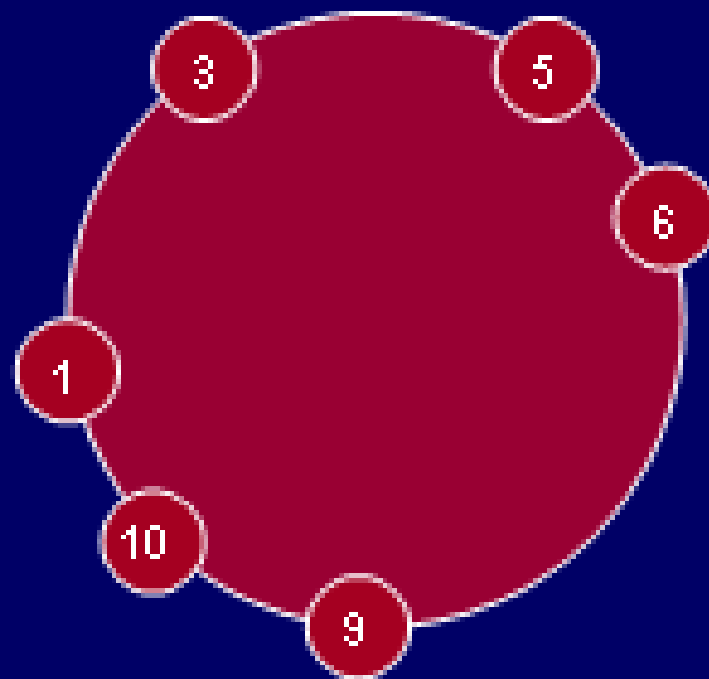


eliminated



Josephus Problem

- $N=10, M=3$

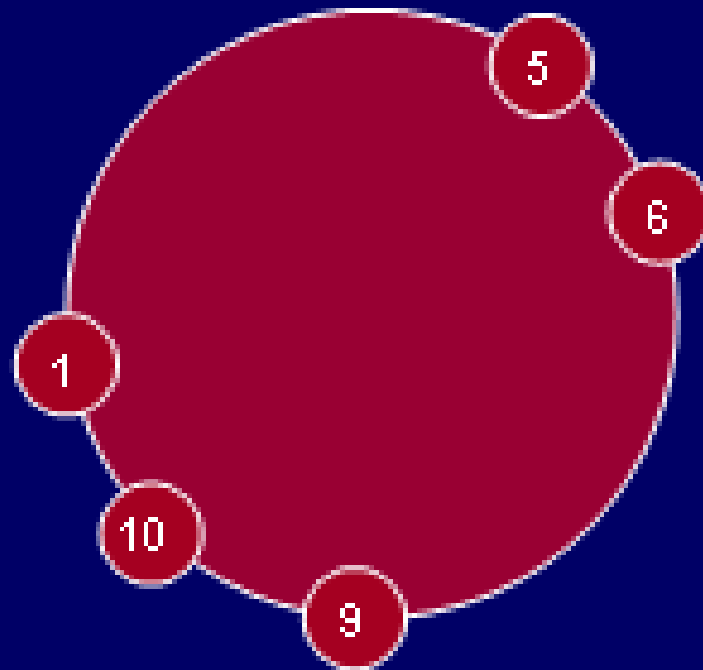


eliminated

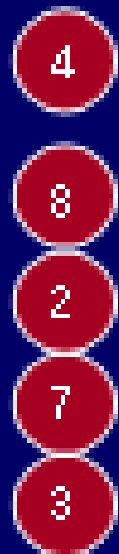


Josephus Problem

- $N=10, M=3$

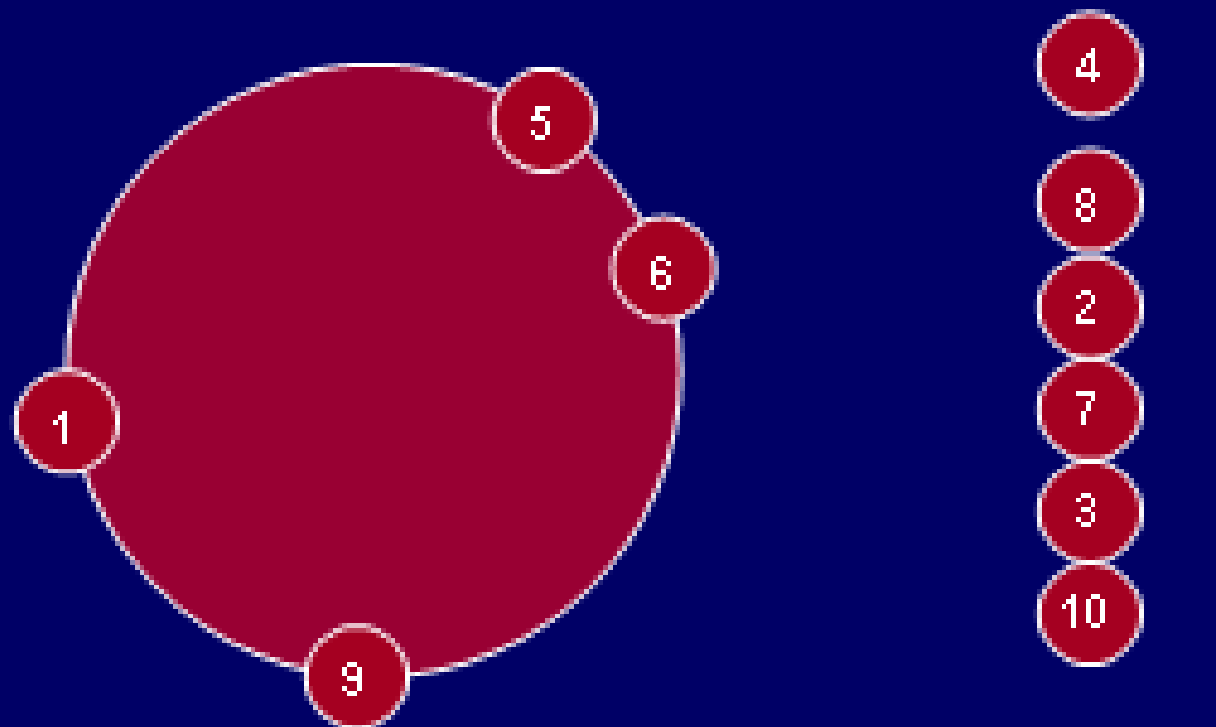


eliminated



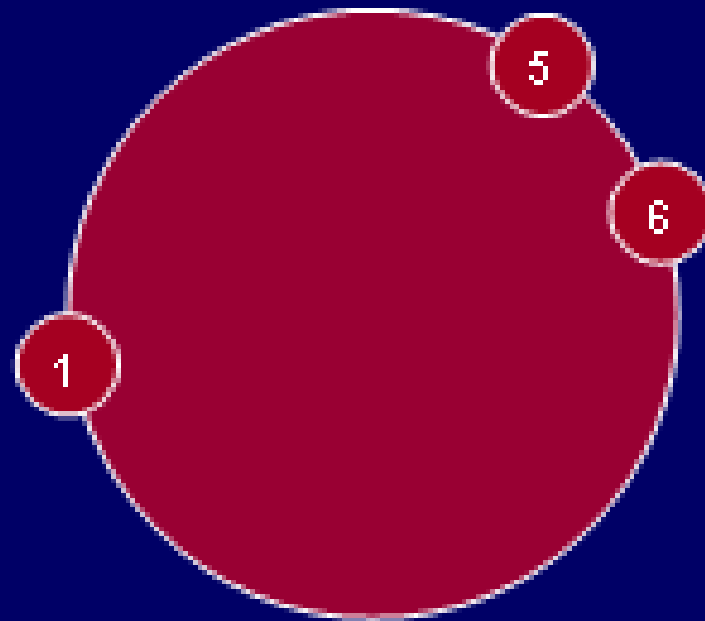
Josephus Problem

- $N=10, M=3$



Josephus Problem

- $N=10, M=3$

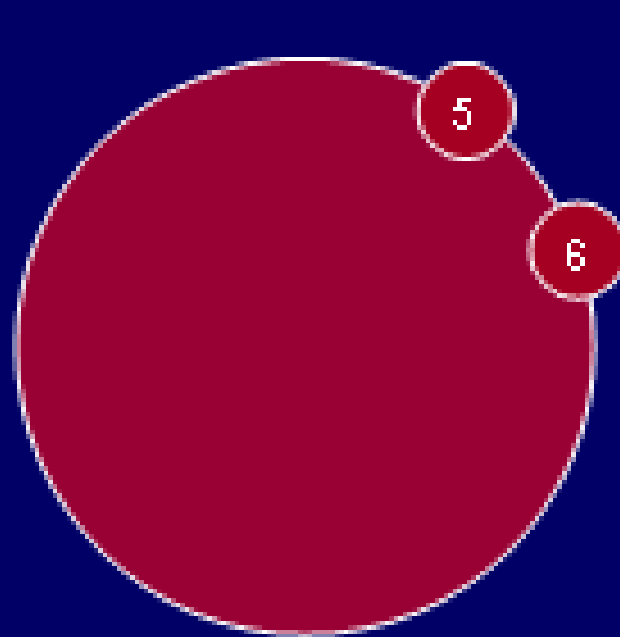


eliminated



Josephus Problem

- $N=10, M=3$

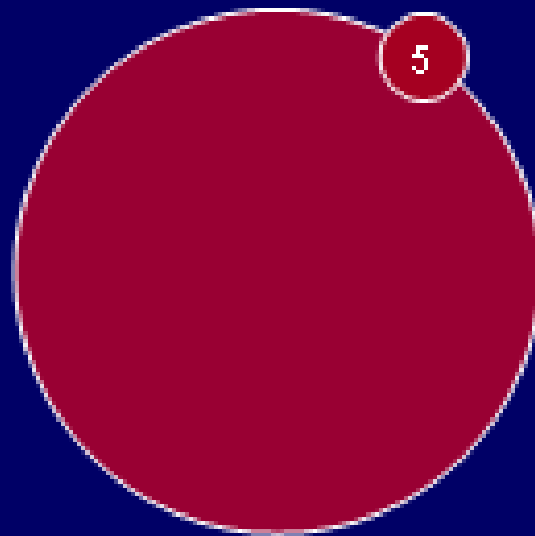


eliminated



Josephus Problem

- $N=10, M=3$



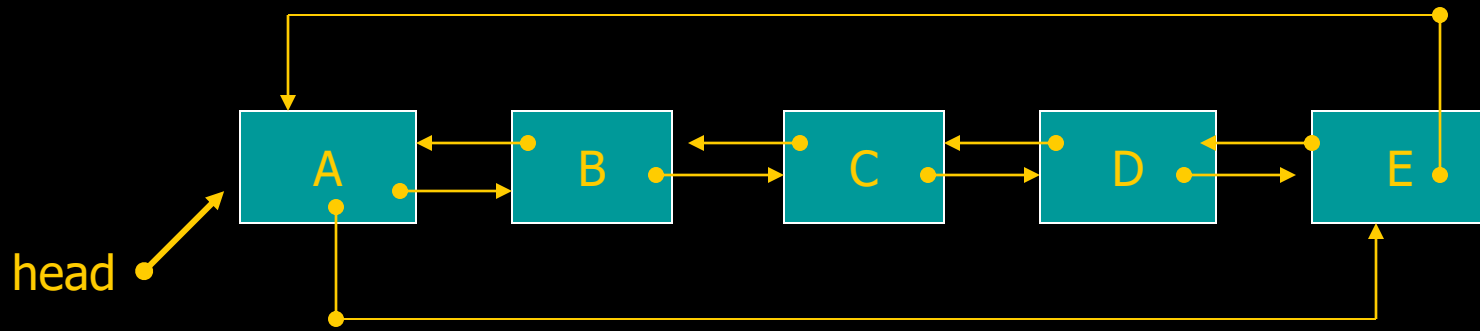
eliminated



Doubly Ring

- **Doubly Ring** data structure can be regarded as **Circular Doubly Link List**.
- In **Circular Doubly Link List**, instead of making last node's **next** pointer to NULL, We assign or point it to **head** node. Similarly, **first** (head) node's **previous** pointer is pointed to last node in list . so that it sounds as **Doubly Ring** data structure.

■ Figure showing Doubly Ring.....



OR

