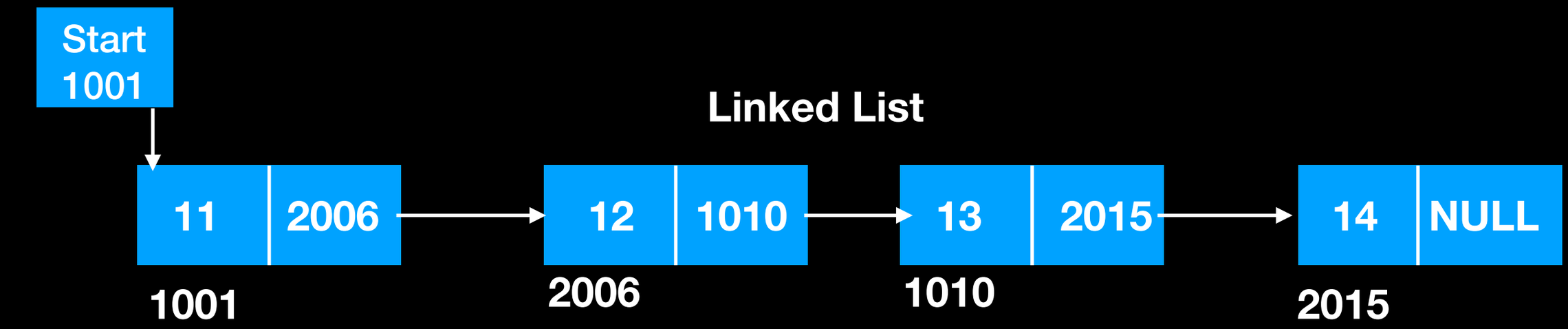
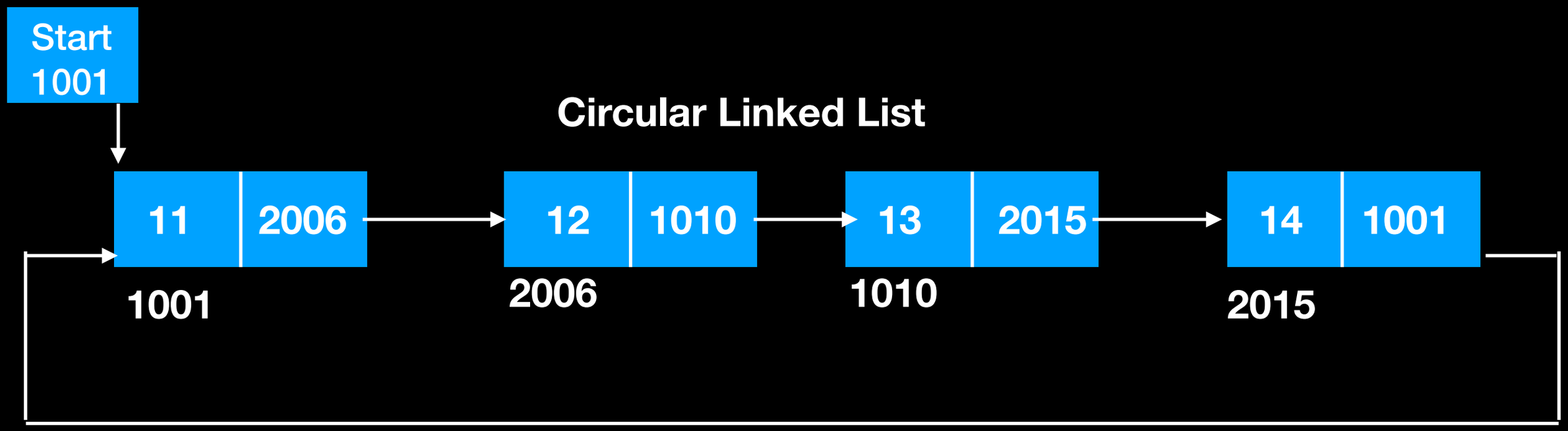
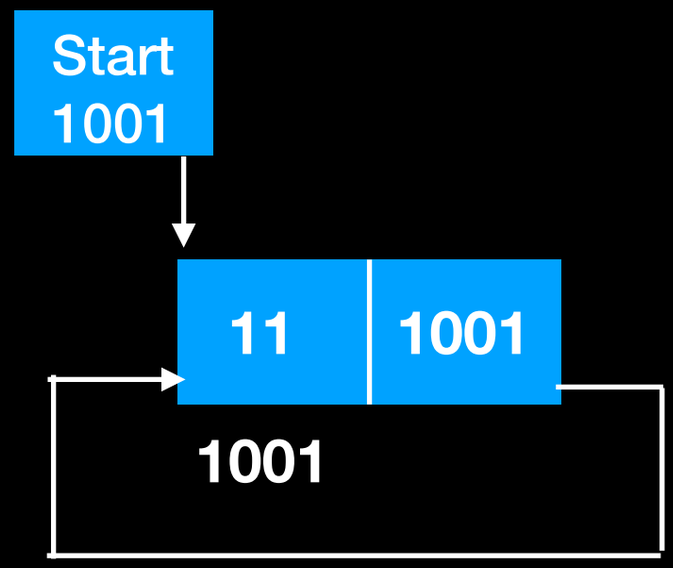
Circular Singly Linked List

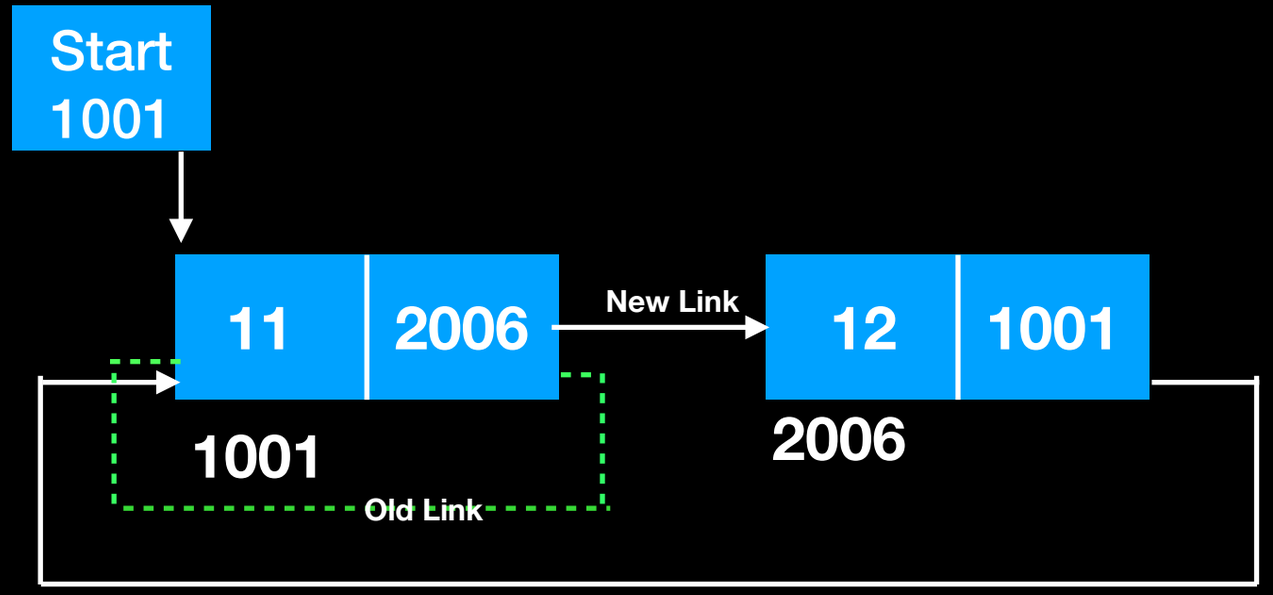
Circular Linked List

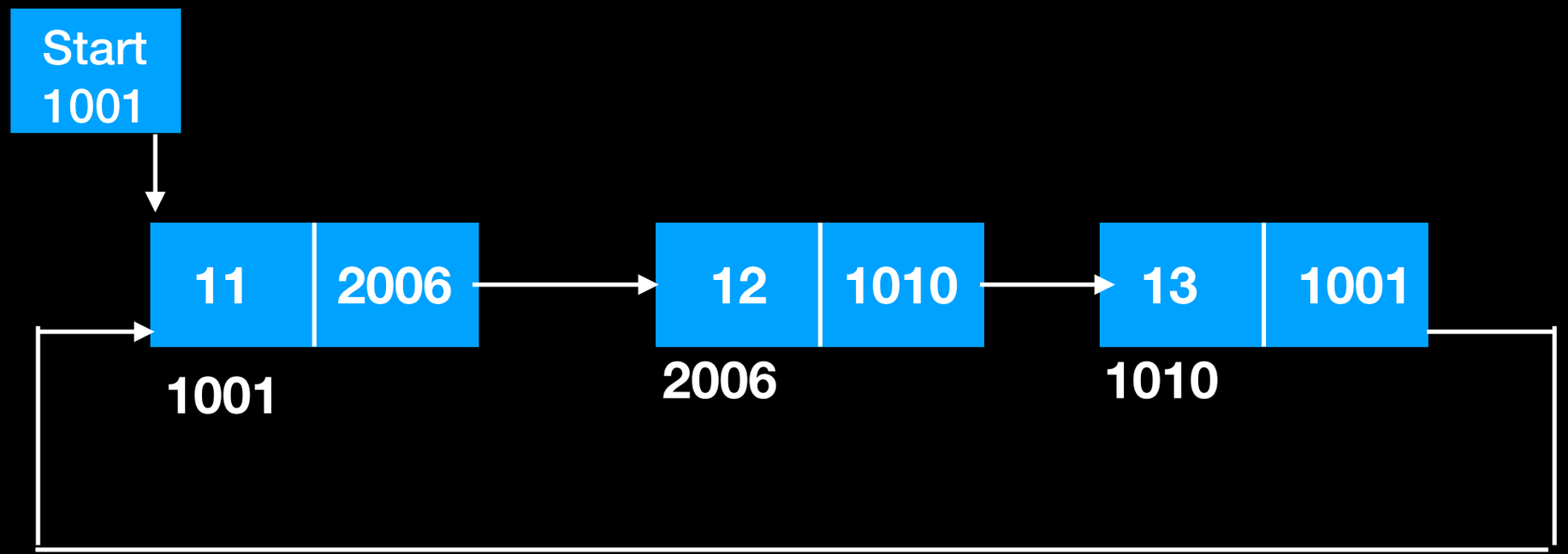
- A Linked List in which last node points to the header node(First Node) is called circular Linked List.
- Last Node of circular linked list does not contain the NULL .

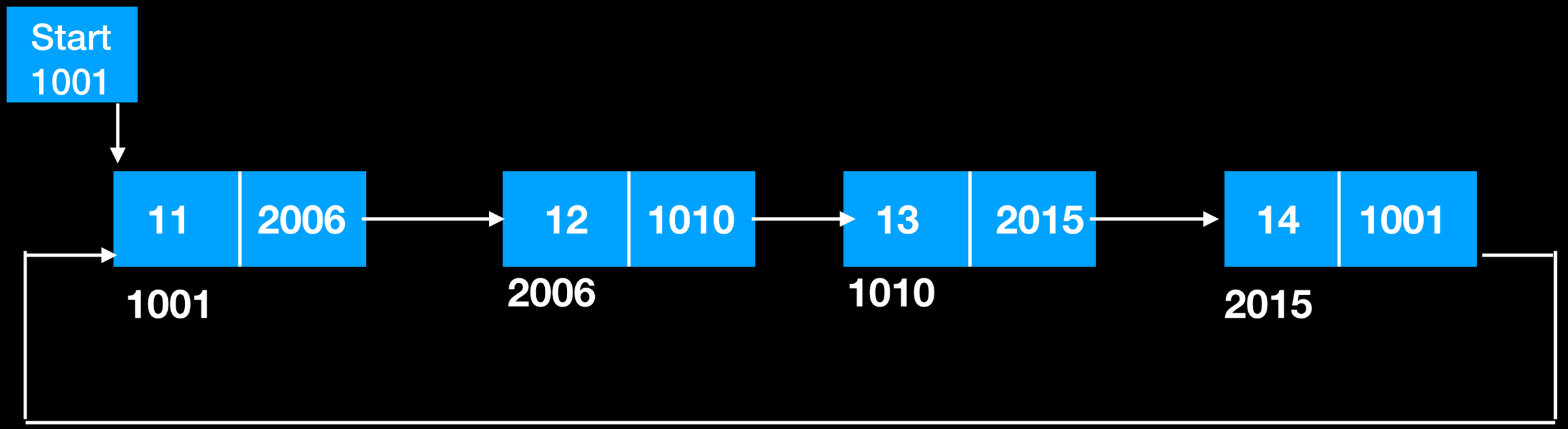

Start
1001
Linked List
11
2006
12
1010
13
2015
14
NULL
1001
2006
1010
2015


Start
1001
Circular Linked List
11
2006
12
1010
13
2015
14
1001
1001
2006
1010
2015


Start
1001
11
1001
1001


Start
1001
11
2006
New Link
12
1001
2006
1001
Old Link

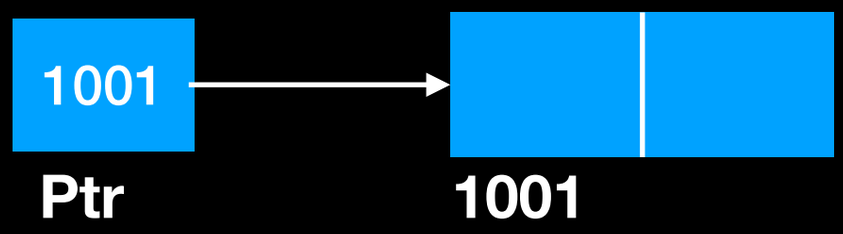

Start
1001
11
2006
12
1010
13
1001
1001
2006
1010


Start
1001
11
2006
12
1010
13
2015
14
1001
1001
2006
1010
2015

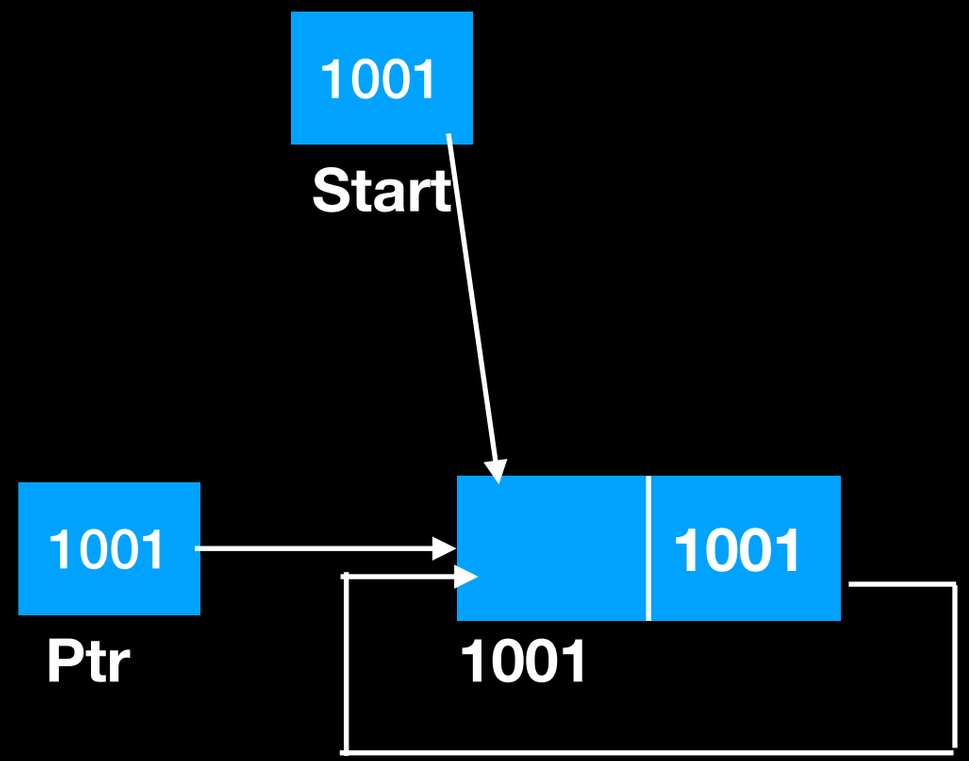
```
Struct node{
        Int data;
             Struct node *next;
          };

   Struct node *start;
```

Start

```
Struct node *ptr, *temp;

 ptr=(struct node*)malloc(sizeof(struct node));
```

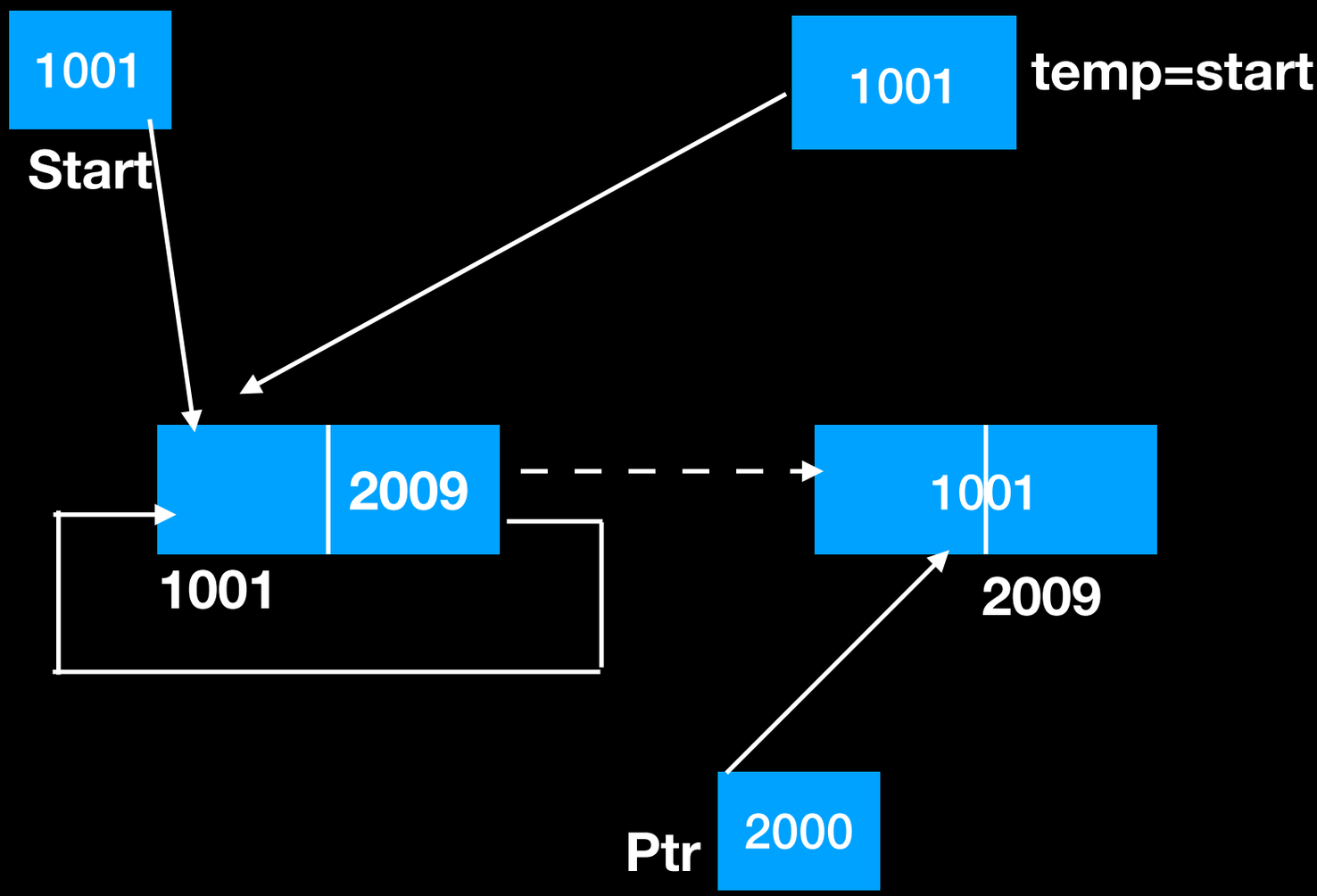


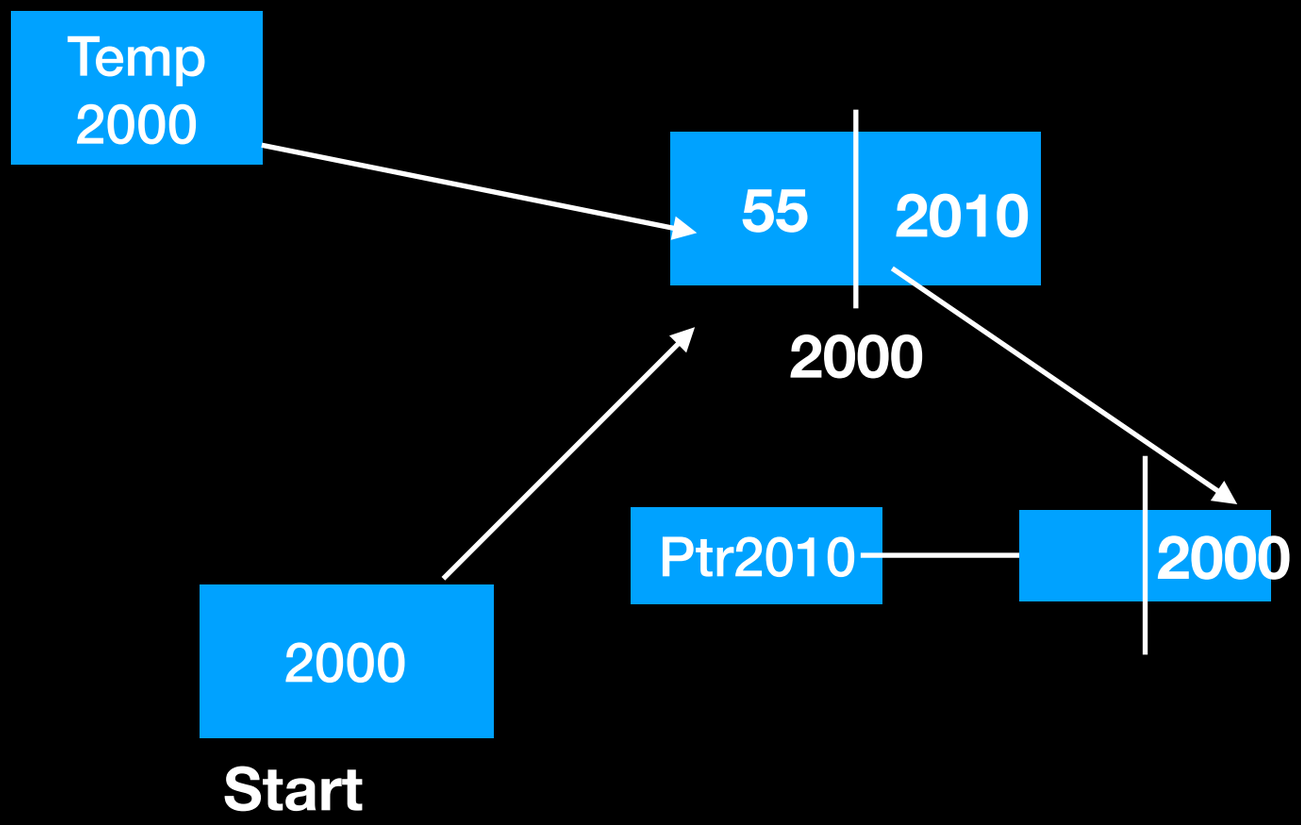
```
if(start==NULL)
{
start=ptr;
ptr->next=start;
}
```



Else

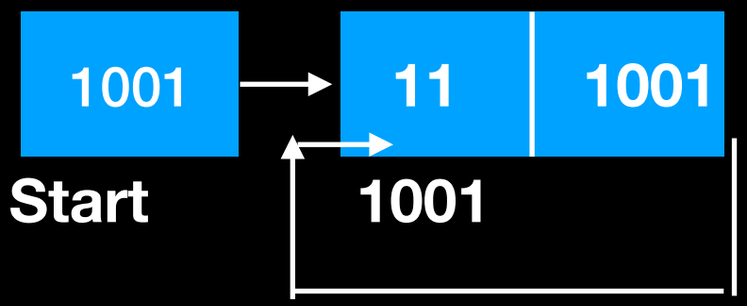
```
temp=start;
while(temp->next!=start)
{
temp=temp->next;
}

temp->next=ptr;
ptr->next=start;
```

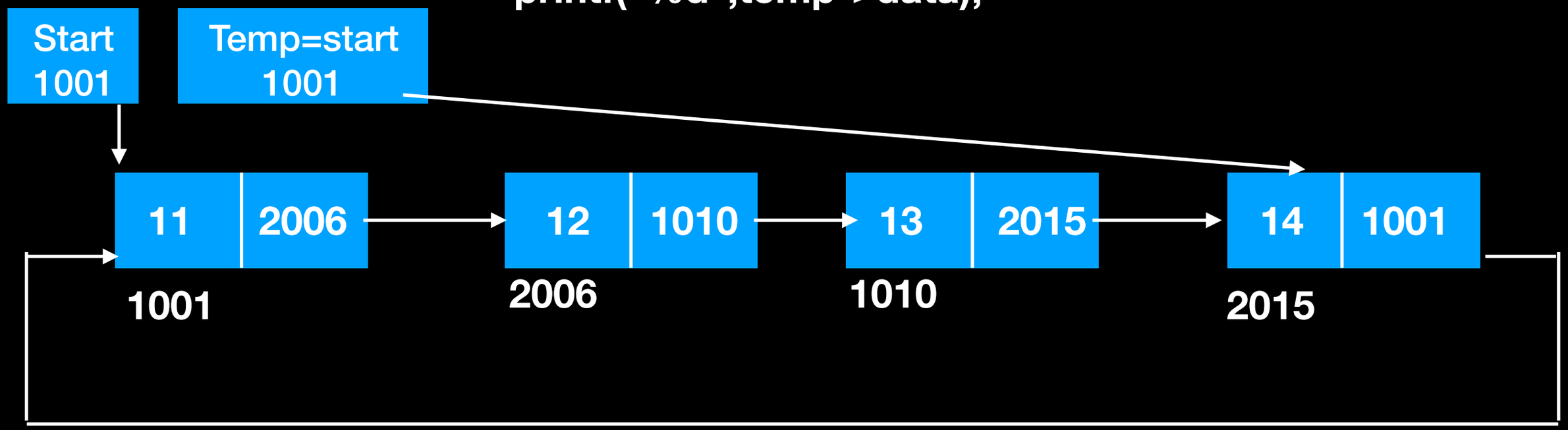



Temp
2000
55
2010
2000
Ptr2010
2000
2000
Start

Print List Data

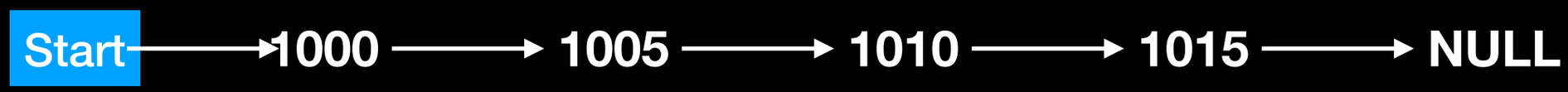
```
Struct node *temp;
temp=start

if(start==NULL)
{
printf("Empty List");
}

Else{
while(temp->next!=start)
{
printf("%d",temp->data);
temp=temp->next;
}
printf("%d",temp->data);
```

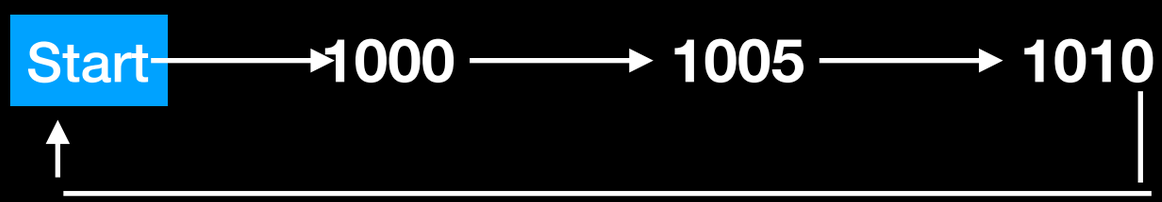




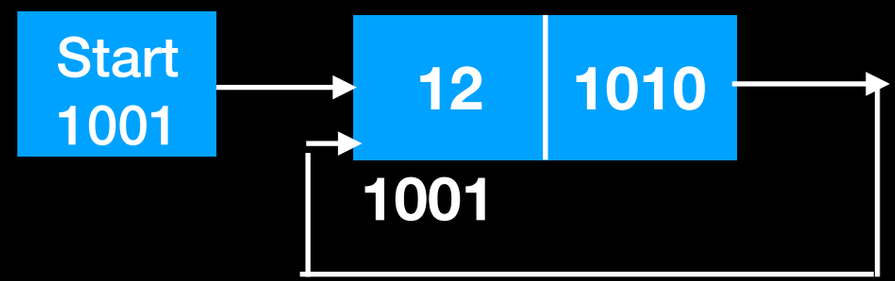
Delete Last Node of Circular Linked List

Delete Last Node

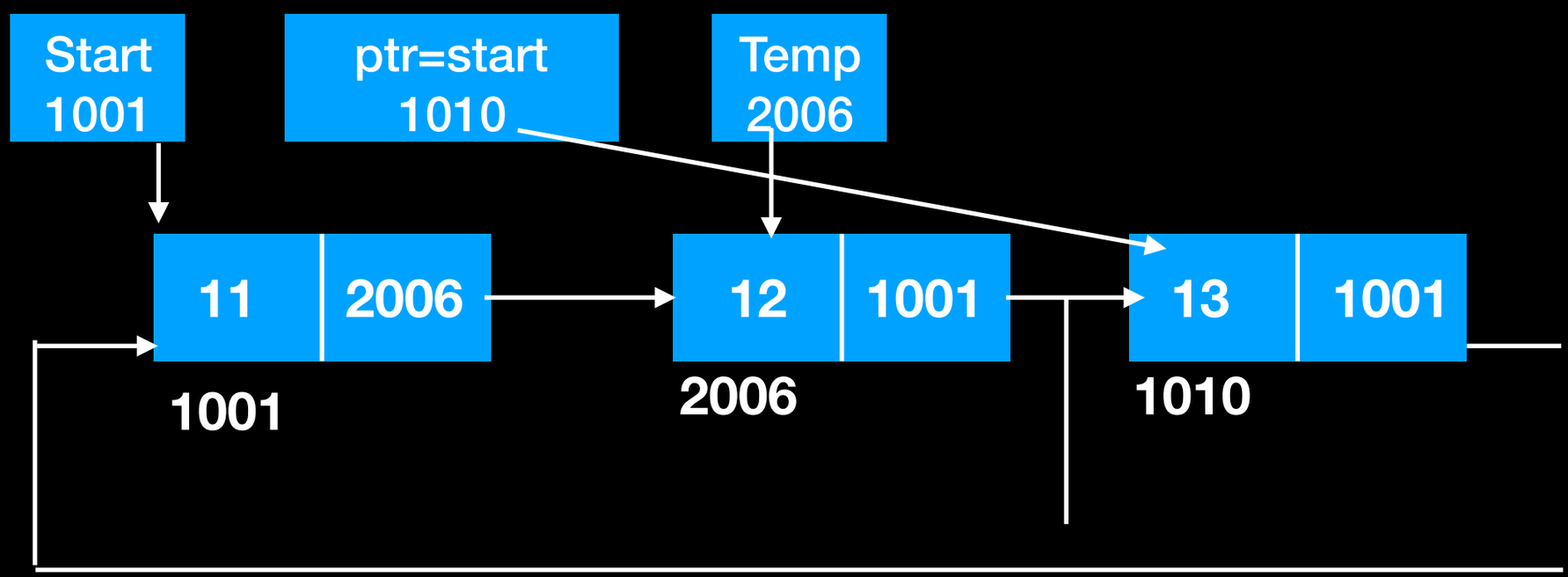




```
if(start->next==start)
                {
            start=NULL
                }
```

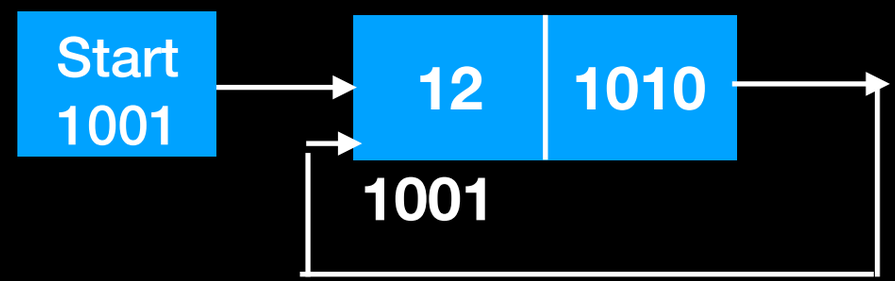


```
ptr=start;
while(ptr->next!=start)
{
temp=ptr;
ptr=ptr->next;
}
temp->next=ptr->next.        // or start
```

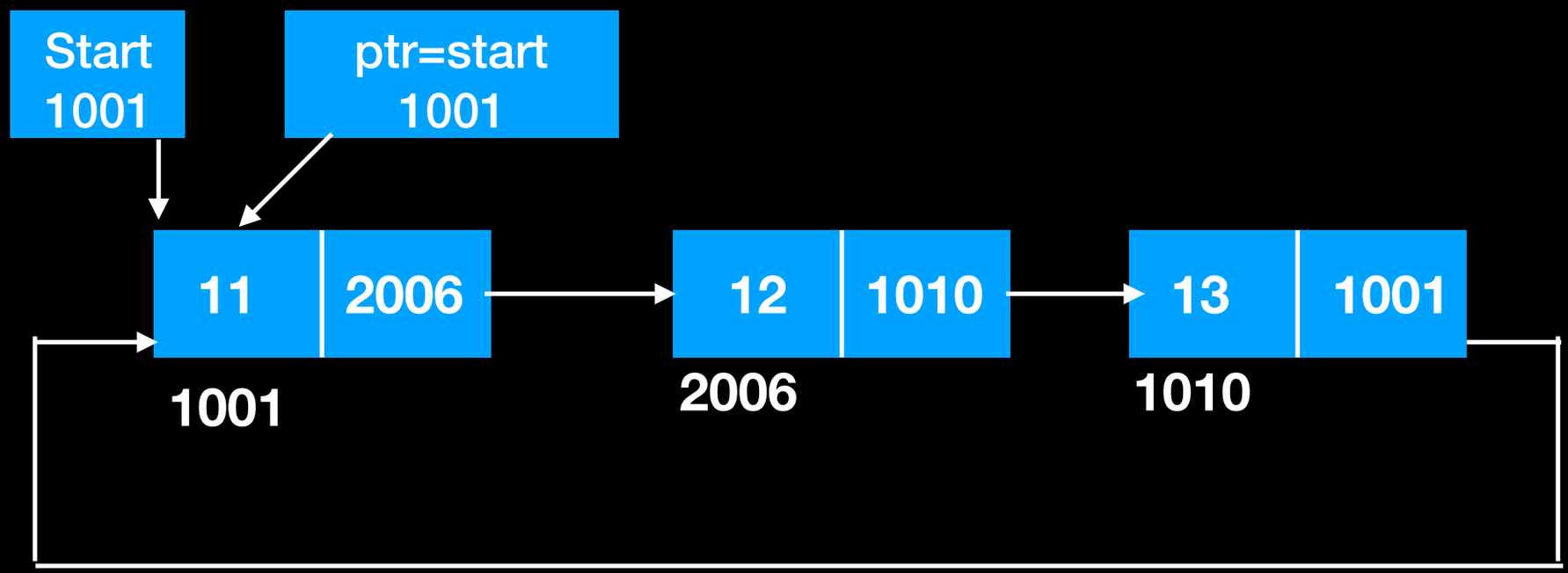


Delete First Node of Circular Linked List

```
if(start->next==start)
                {
            start=NULL
                }
```



```
ptr=start;
while(ptr->next!=start)
{
ptr=ptr->next;
}
ptr->next=start->next;
start=ptr->next;
```



Search data in circular Linked List

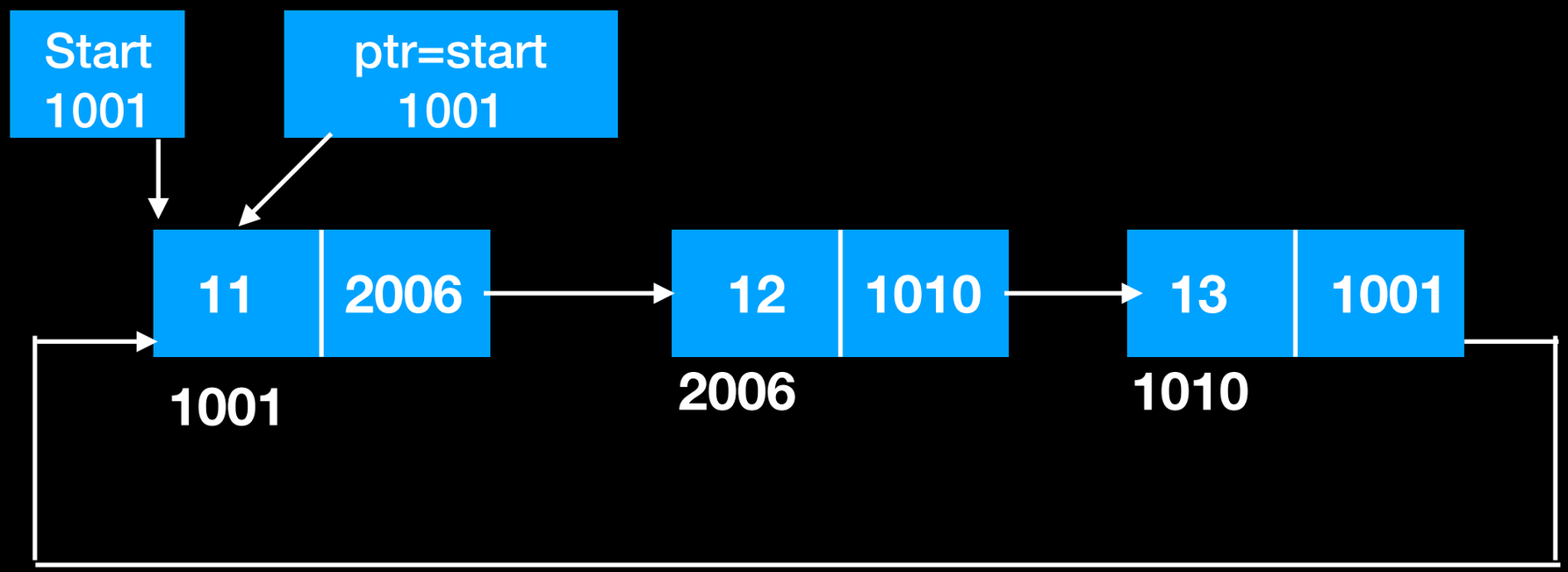
ptr=start
if(ptr==null)
printf(“empty List”)

start=NULL

ptr=NULL

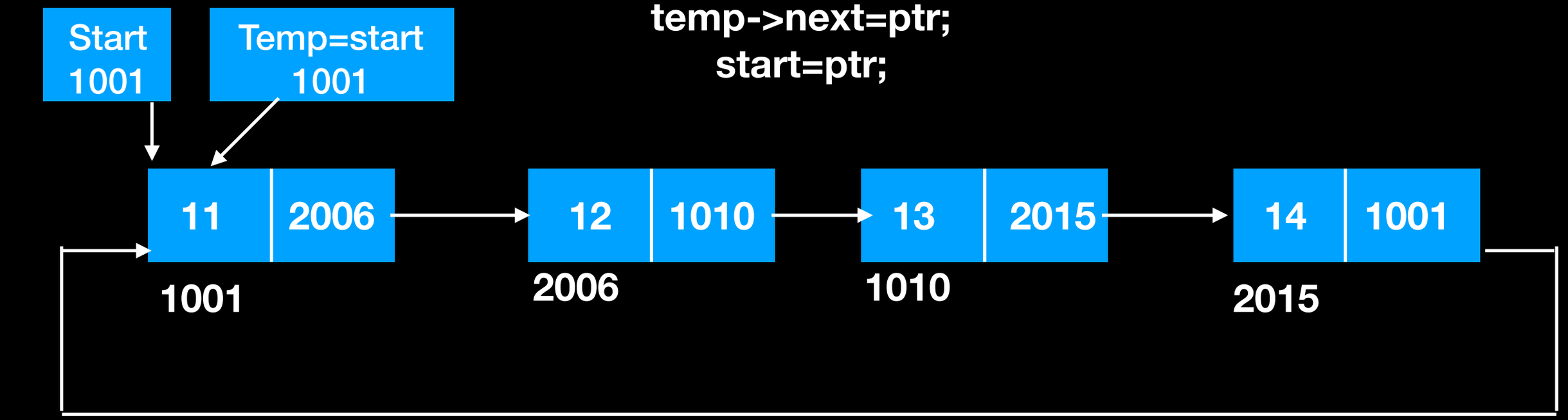
Search data in circular Linked List

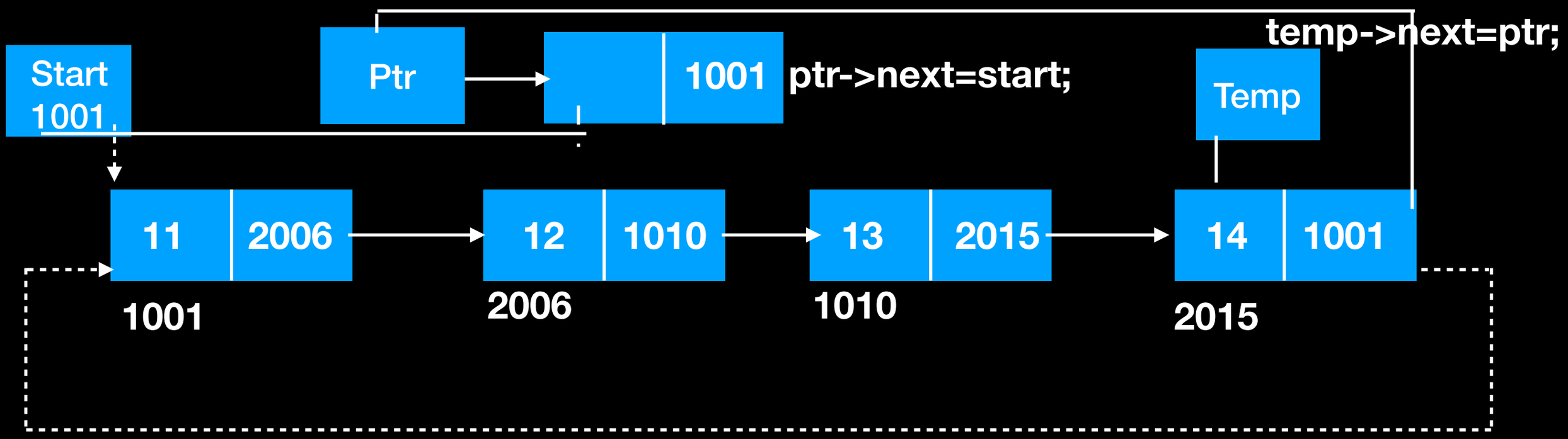
```
if(start->data==item)
printf()
else(
while(ptr->next!=start)
if(ptr->data==item)
{
Print and break;
}
}
```



Node insertion at beginning

```
temp=start;
while(temp->next!=start)
{
temp=temp->next;
}
ptr->next=start;
temp->next=ptr;
start=ptr;
```





Circular Linked List Program

```c
#include <stdio.h>
#include <stdlib.h>
void createList(void);
void display(void);
void deleteNode(void);
struct node{
    int data;
    struct node* next;
};

struct node* start=NULL;

int main(int argc, const char * argv[]) {

    for(int i=0;i<5;i++){
        createList();
    }

    display();
    deleteNode();
    deleteNode();
    display();
    return 0;
}
```

```
void createList()
{
    struct node *ptr,*temp;
    int data1;
    ptr=(struct node*)malloc(sizeof(struct node));
    printf("Enter Data\n");
    scanf("%d",&data1);
    ptr->data=data1;
    if(start==NULL)
    {
        start=ptr;
        ptr->next=start;
    }
    else{
        temp=start;
        while(temp->next!=start){
            temp=temp->next;
        }
        temp->next=ptr;
        ptr->next=start;
    }

}
```

```
void display(){

    struct node *ptr;
    ptr=start;

    if(start==NULL){
        printf("Empty List");
    }
    else{
        while(ptr->next!=start)
        {
            printf("%d\n",ptr->data);
            ptr=ptr->next;
        }
        printf("%d\n",ptr->data);
    }
}
```

```
void deleteNode(){
    struct node *ptr,*temp = NULL;
    if(start->next==start)
    {
        start=NULL;
        printf("node Deleted\n");
    }
    else{
        ptr=start;
        while(ptr->next!=start){
            temp=ptr;
            ptr=ptr->next;
        }
        temp->next=ptr->next;
        printf("Node Deleted");
    }

}
```