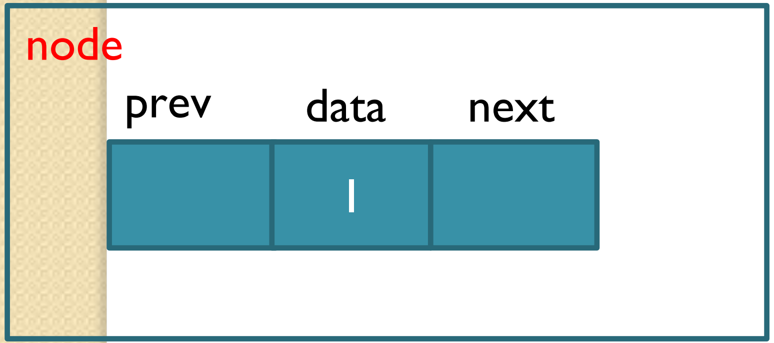
Double Linked List

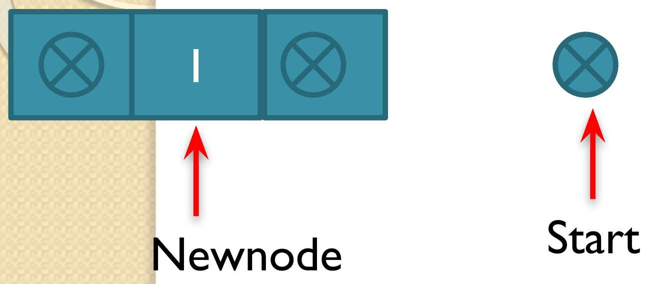
A doubly linked list is a more complex type of linked list which contains a pointer to the next as well as the previous node in the sequence



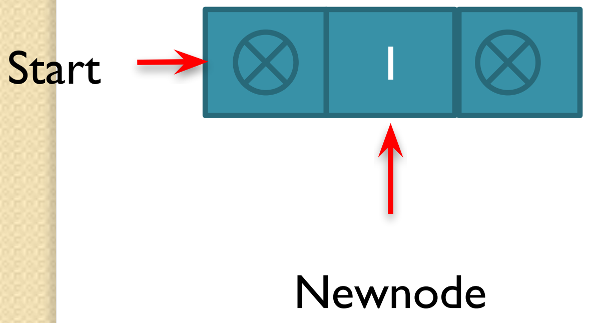
```
struct node
{   struct node *prev;
    int data;
    struct node *next;
};
```

Double Linked List: Creation

STEP1: Create Newnode with data value



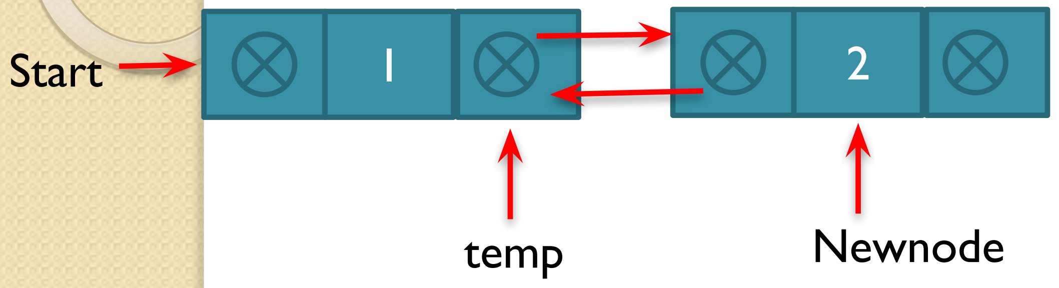
STEP2: if Start is NULL then Assign Newnode to Start



```
if(Start==NULL)
     Start= Newnode;
else{
     temp= Start ;
while(temp->next!=NULL)
            temp=temp->next;
     temp->next=Newnode;
Newnode->prev=temp;
}
```

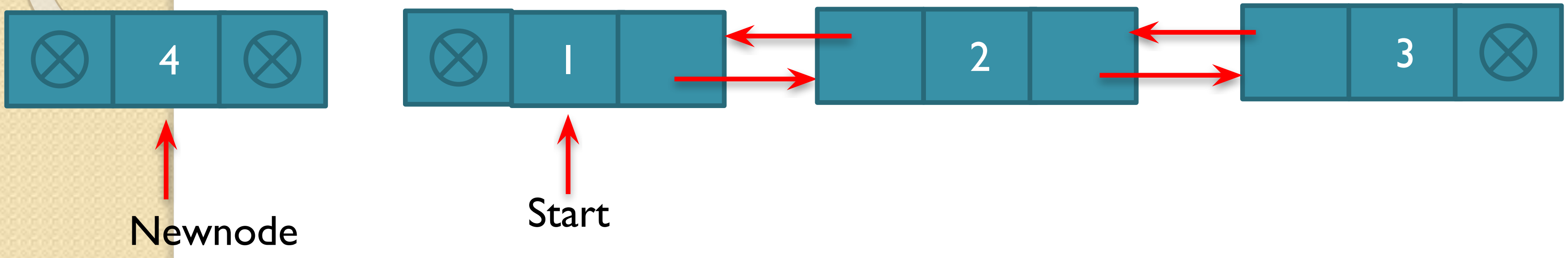
Double Linked List: Creation

CREATE REMAINING NODES



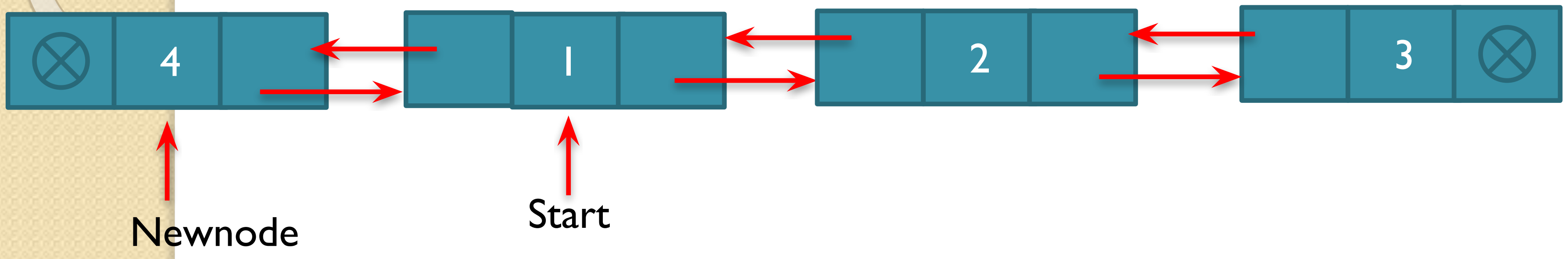
```
if(start==NULL)
        Start=temp;
else{
        temp=Start;
while(temp->next!=NULL)
                temp=temp->next;
        temp->next=Newnode;
        Newnode->prev=temp;
}
```

Double Linked List
Insertion: At Start Position



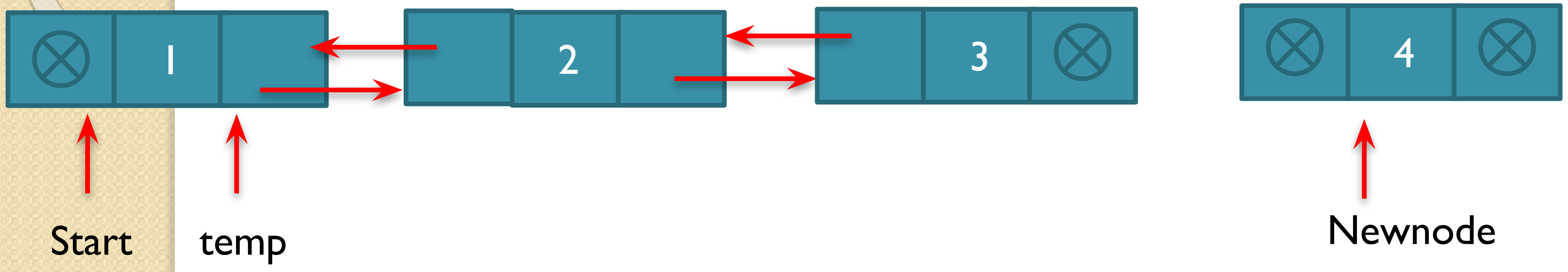
```
Newnode ->next=Start;
Start->prev= Newnode;
Start= Newnode;
```

Double Linked List
Insertion: At Start Position



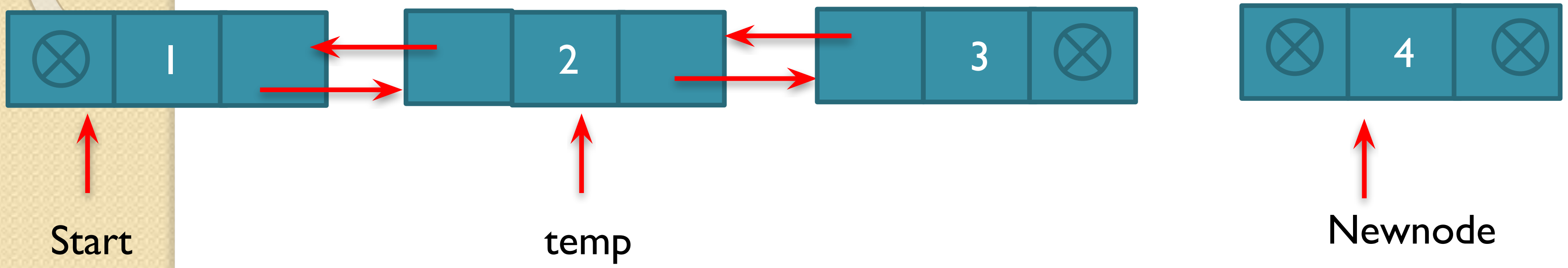
```
Newnode ->next=Start;
Start->prev= Newnode;
Start= Newnode;
```

Double Linked List
Insertion: At Last Position



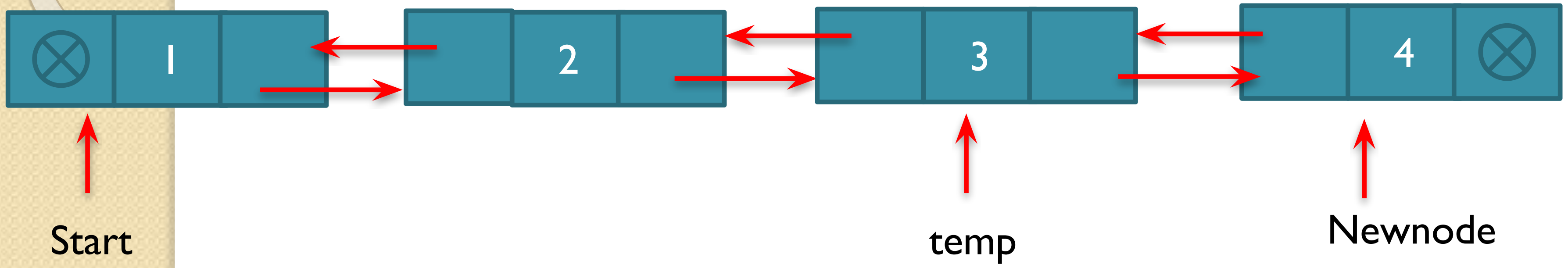
```
temp=Start;
while(temp->next!=NULL)
    temp=temp->next;
temp->next= Newnode;
Newnode ->prev=temp;
```

Double Linked List
Insertion: At Last Position



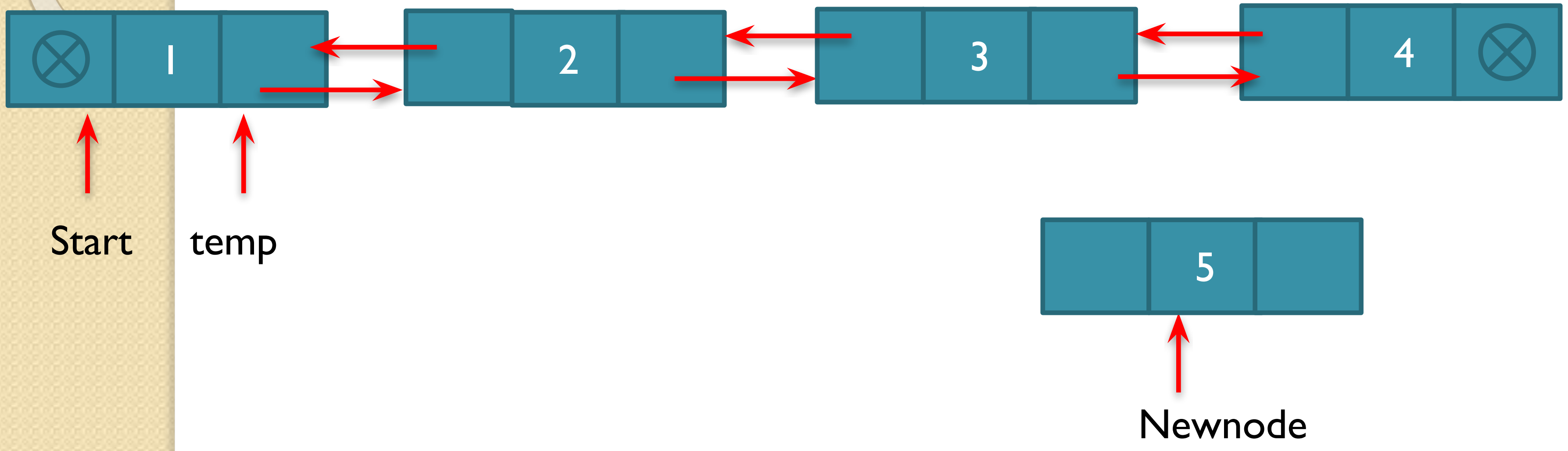
```
temp=Start;
while(temp->next!=NULL)
    temp=temp->next;
temp->next= Newnode;
Newnode ->prev=temp;
```

Double Linked List
Insertion: At Last Position



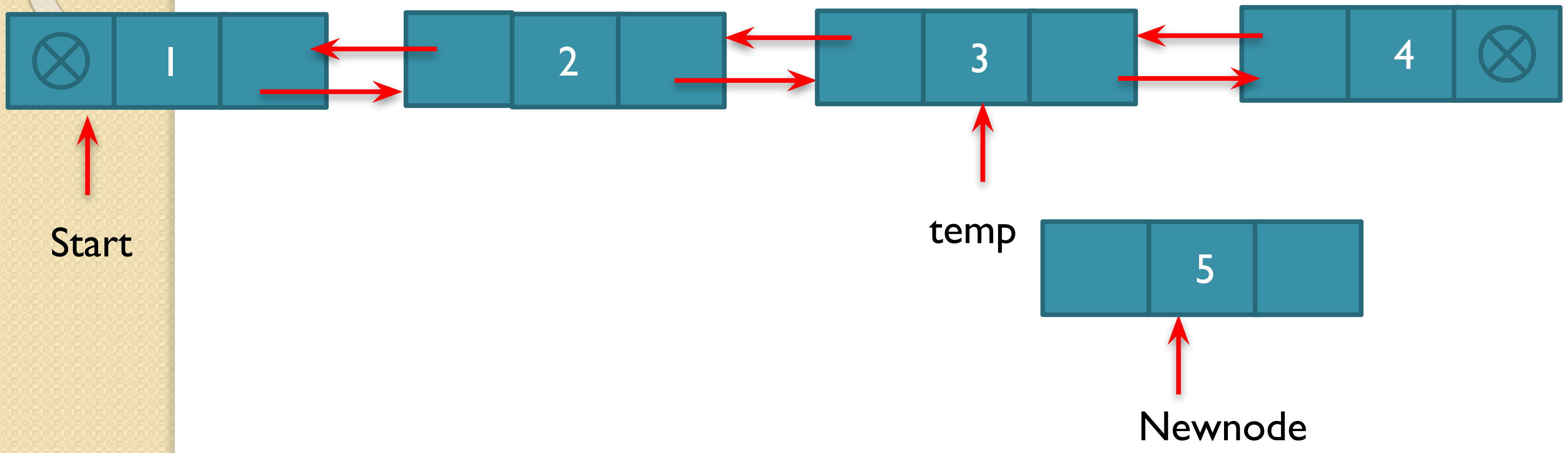
```
temp=Start;
while(temp->next!=NULL)
    temp=temp->next;
temp->next= Newnode;
Newnode ->prev=temp;
```

Double Linked List
Insertion: In Between



Step1: Enter a position and move temp pointer reach to position - 1.

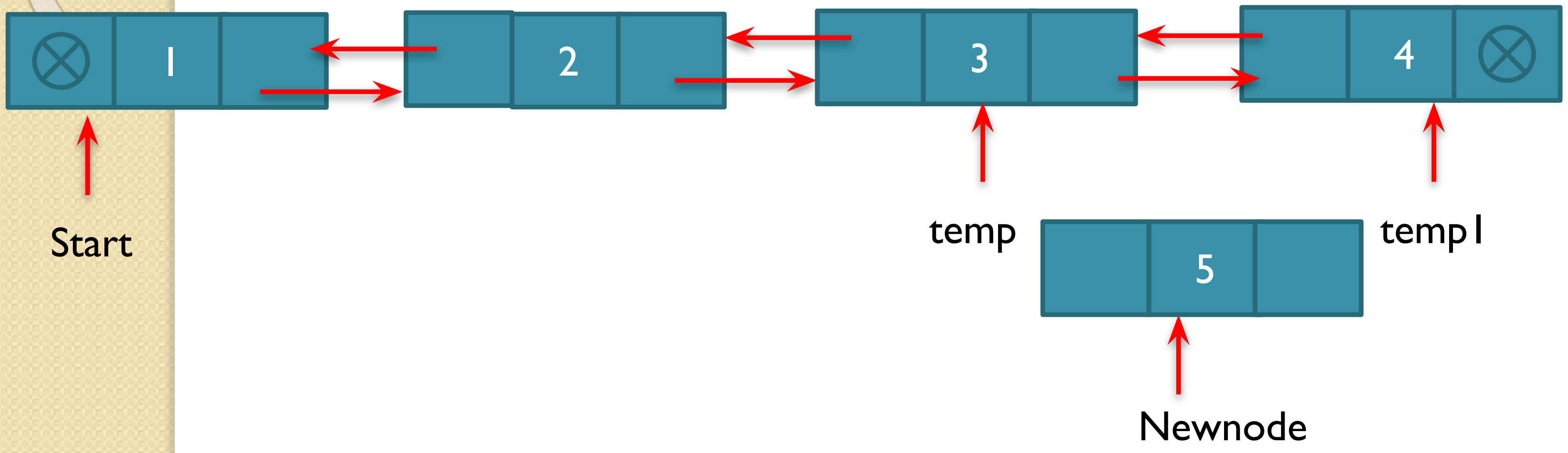
Double Linked List
Insertion: In Between



Step1: Enter a position and move temp pointer reach to position - 1.

Step2: check for the correctness of temp, if correct follow the steps below:

Double Linked List
Insertion: In Between

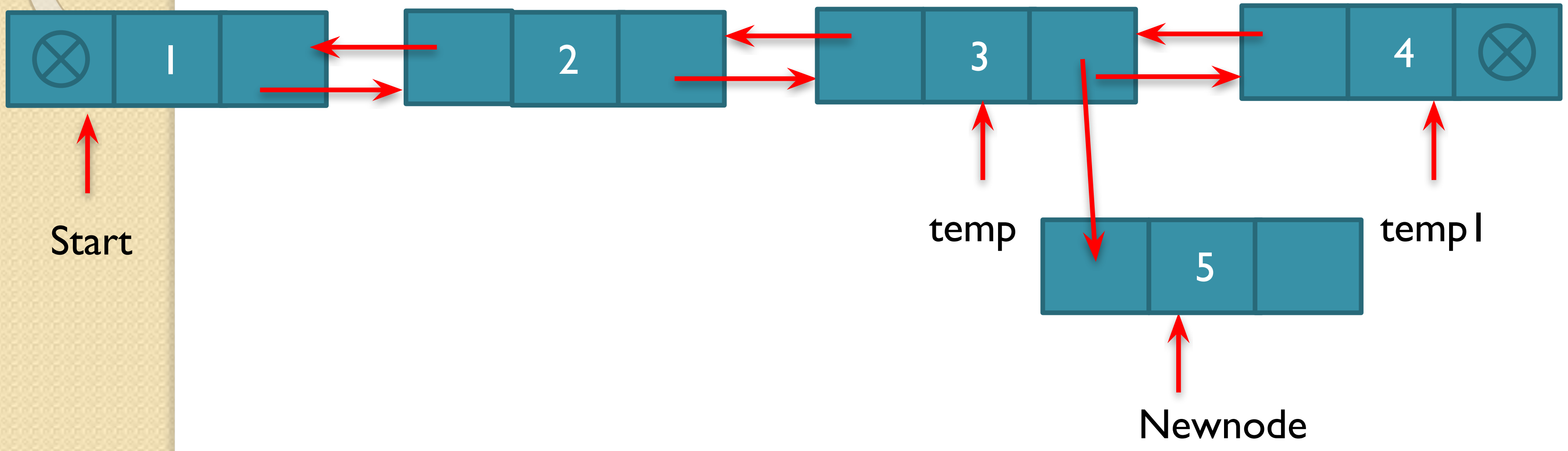


Step1: Enter a position and move temp pointer reach to position - 1.

Step2: check for the correctness of temp, if correct follow the steps below:

Step3: temp1=temp->next

Double Linked List
Insertion: In Between



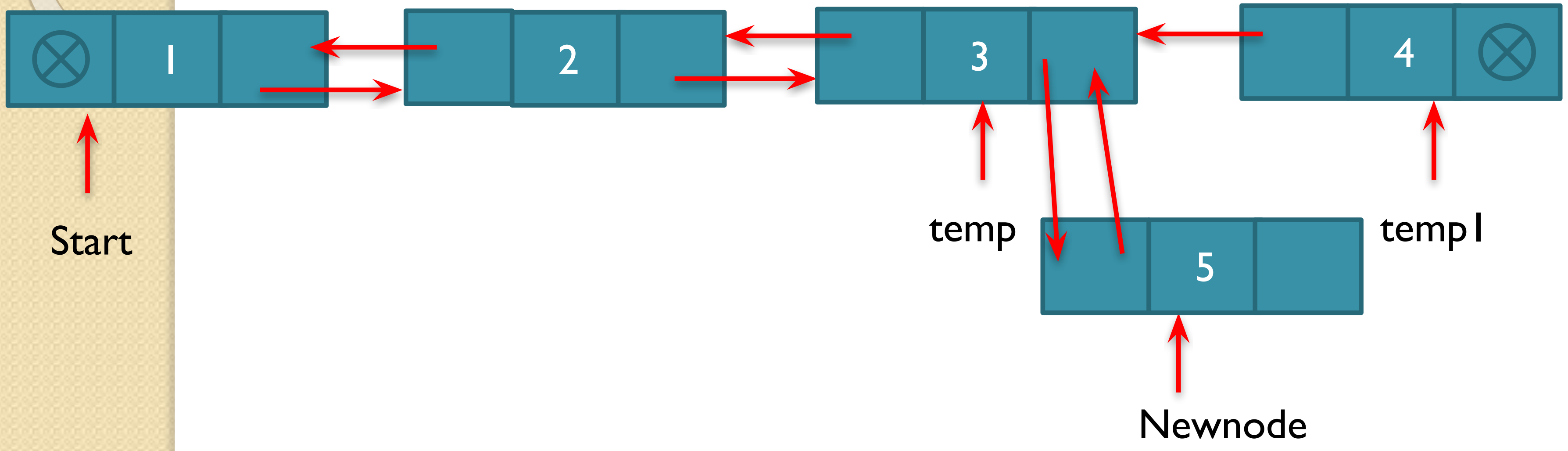
Step1: Enter a position and move temp pointer reach to position - 1.

Step2: check for the correctness of temp, if correct follow the steps below:

Step3: temp1=temp->next

Step4: i) temp->next= Newnode

Double Linked List
Insertion: In Between



Step1: Enter a position and move temp pointer reach to position - 1.

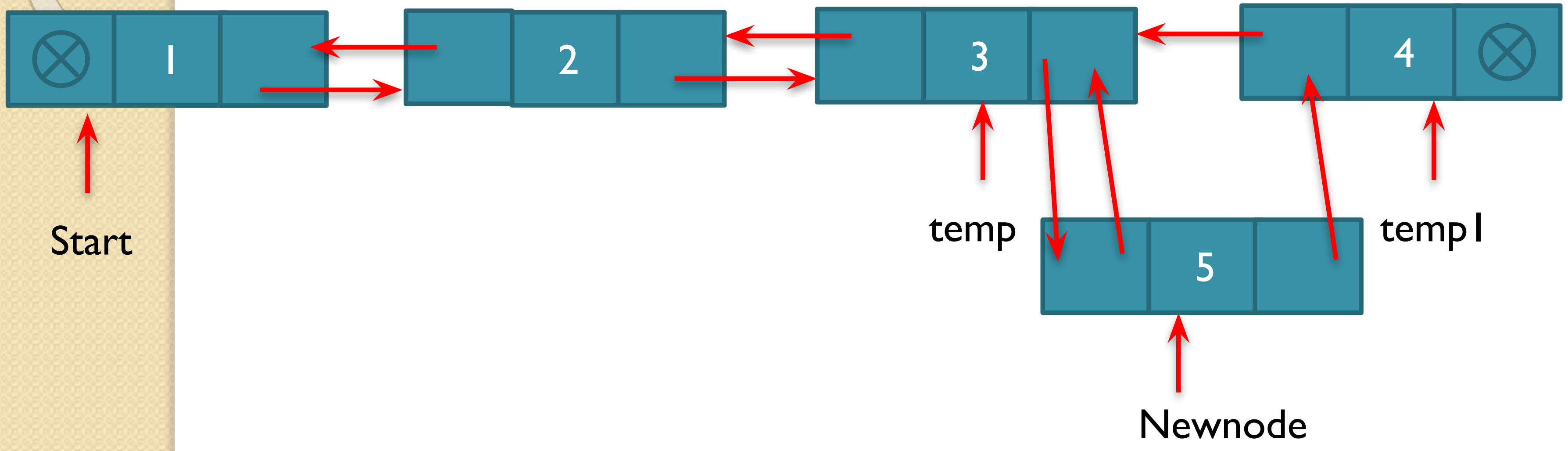
Step2: check for the correctness of temp, if correct follow the steps below:

Step3: temp1=temp->next

Step4: i) temp->next=temp->prev

ii) Newnode ->prev=temp

Double Linked List
Insertion: In Between



Step1: Enter a position and move temp pointer reach to position - 1.

Step2: check for the correctness of temp, if correct follow the steps below:

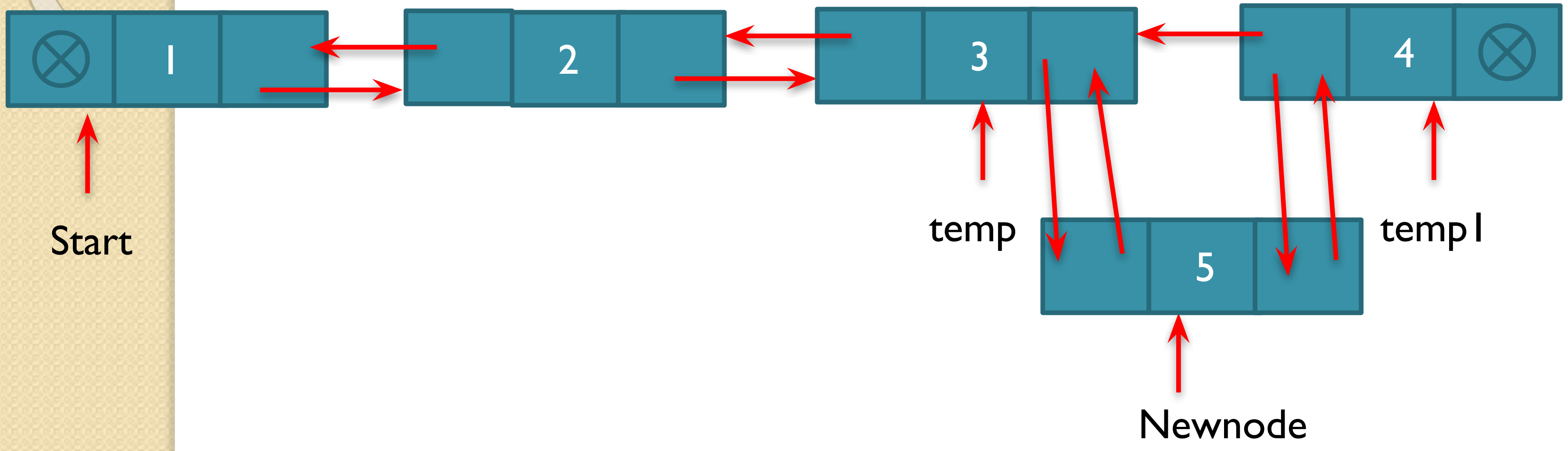
Step3: temp1=temp->next

Step4: i) temp->next=temp->prev

ii) temp->prev=temp

iii) Newnode ->next=temp1

Double Linked List
Insertion: In Between



Step1: Enter a position and move temp pointer reach to position - 1.

Step2: check for the correctness of temp, if correct follow the steps below:

Step3: temp1=temp->next

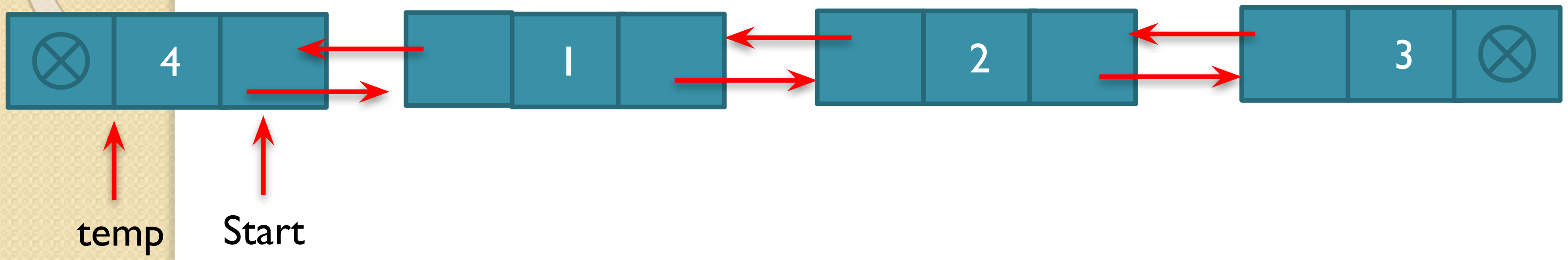
Step4: i) temp->next=temp->prev

ii) temp->prev=temp

iii) temp->next=temp1

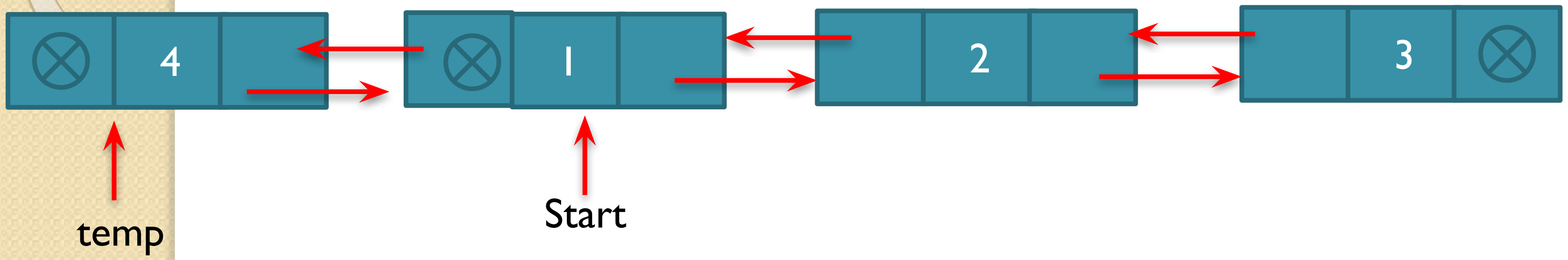
iv) temp1->prev= Newnode

Double Linked List
Delete node at Start Position



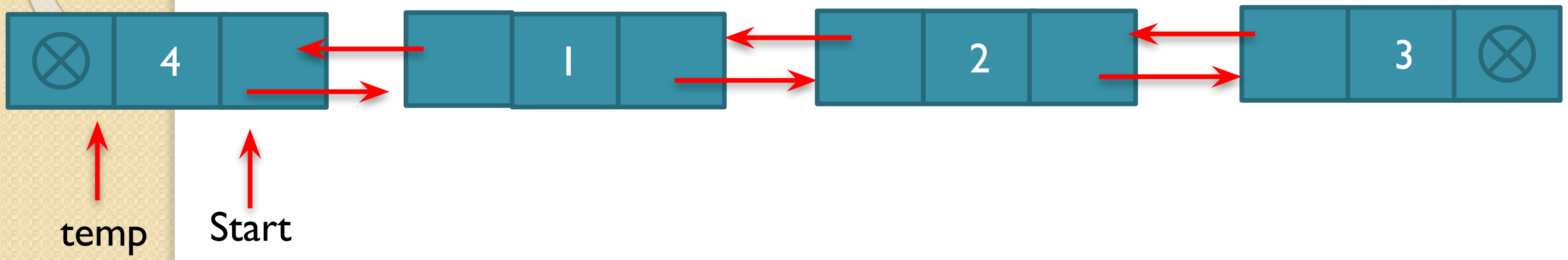
```
temp=Start;
Start=Start->next;
free(temp);
Start->prev=NULL;
```

Double Linked List
Delete node at Start Position



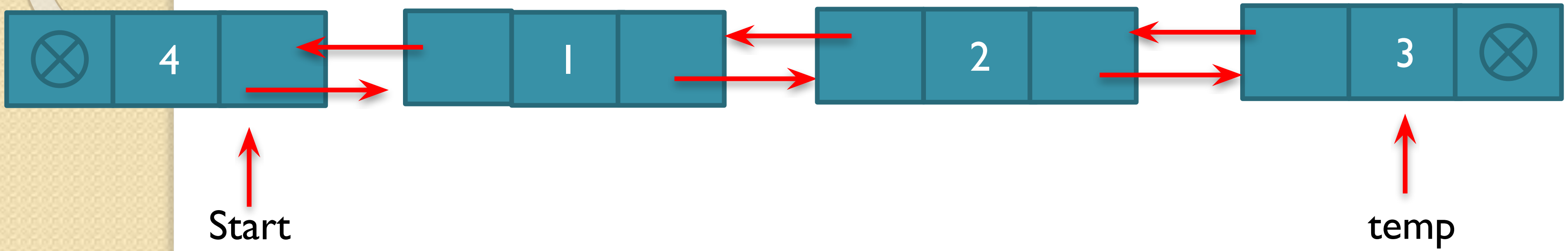
```
temp=Start;
Start=Start->next;
free(temp);
Start->prev=NULL;
```

Double Linked List
Delete node at Last Position



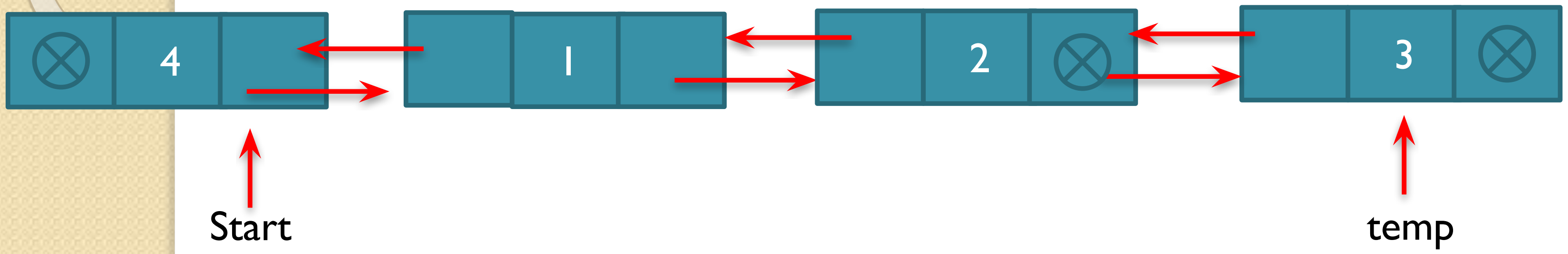
```
temp=Start;
while(temp->next!=NULL)
      temp=temp->next;
temp->prev->next=NULL;
free(temp);
```

Double Linked List
Delete node at Last Position



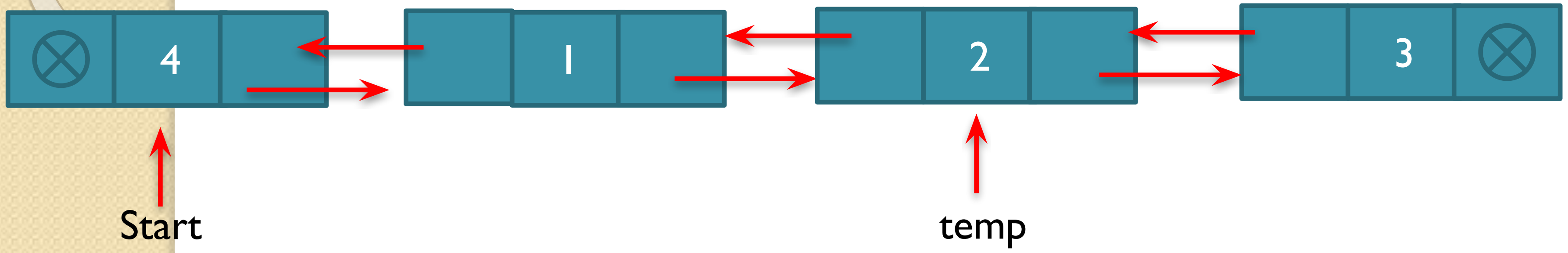
```
temp=Start;
while(temp->next!=NULL)
    temp=temp->next;
temp->prev->next=NULL;
free(temp);
```

Double Linked List
Delete node at Last Position

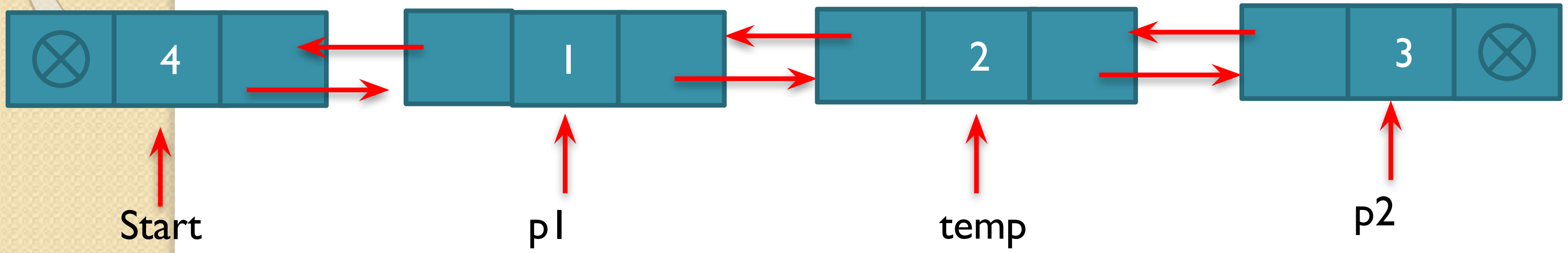


```
temp=Start;
while(temp->next!=NULL)
      temp=temp->next;
temp->prev->next=NULL;
free(temp);
```

Double Linked List
Delete node at some in between position

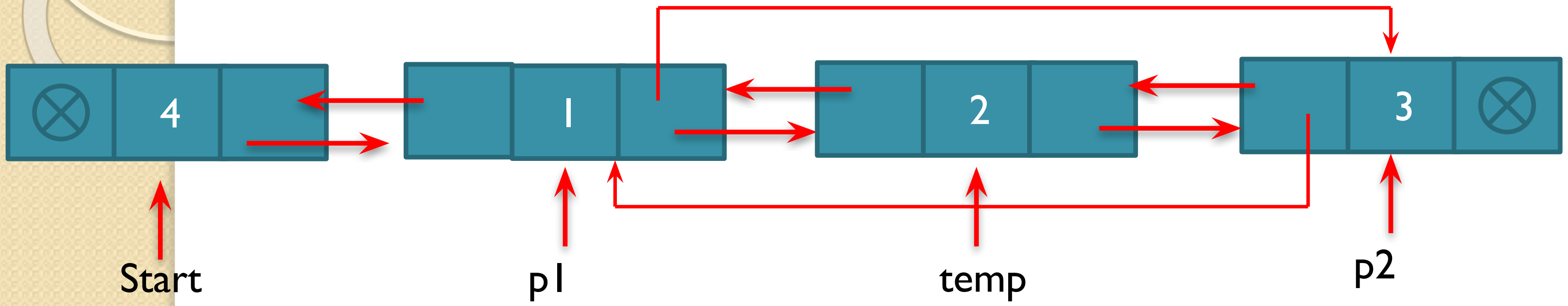


Double Linked List
Delete node at some in between position



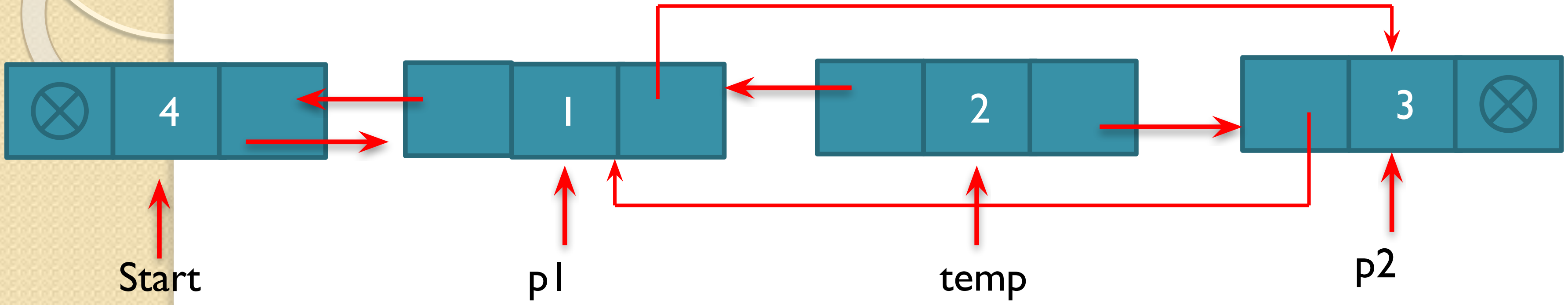
```
p1=temp->prev;
p2=temp->next;
p1->next=p2;
p2->prev=p1;
```

Double Linked List
Delete node at some in between position



```
p1=temp->prev;
p2=temp->next;
p1->next=p2;
p2->prev=p1;
```

Double Linked List
Delete node at some in between position



```
p1=temp->prev;
p2=temp->next;
p1->next=p2;
p2->prev=p1;
free(temp);
```