

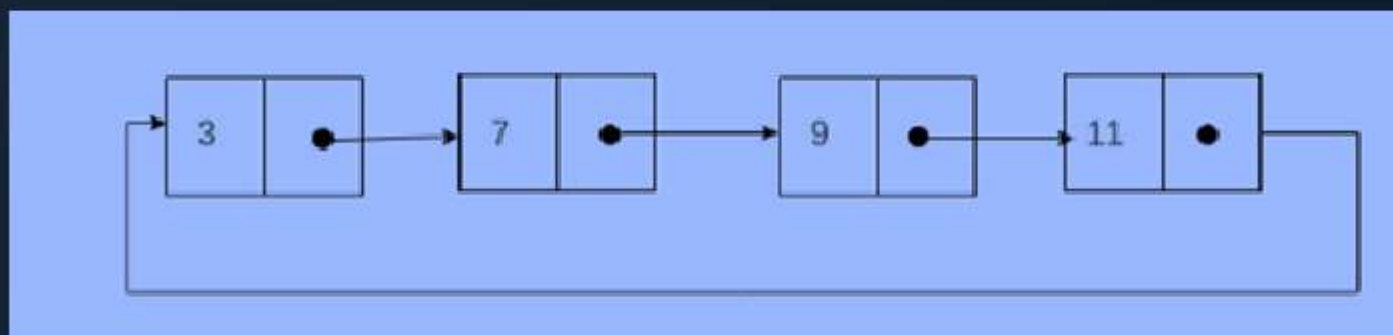
Circular List

Q3. What is circular link list describe its functions: insert(), append() search() delete()

A circular linked list is a type of linked list in which the last node points to the first node, forming a circular loop. This means that there is no beginning or end to the linked list, and it can be traversed in either direction.

Circular linked lists are often used to implement data structures such as queues and ring buffers. They are also used in some operating systems to implement task scheduling and memory management.

Example:



```
#include <iostream>
using namespace std;
struct node
{
    int data;
    node *link;
};
class cirlist
{
private:
    node *start, *cur, *temp;
public:
    cirlist()
    {
        start = NULL;
    }
}
```

```
void append(int n)
{
    if (start == NULL)
    {
        start = new node;
        start->data = n;
        start->link = start;
    }
    else
    {
        cur = start;
        while (cur->link != start)
            cur = cur->link;
        temp = new node;
        temp->data = n;
        temp->link = start;
        cur->link = temp;
    }
    cout<<n<<" inserted in list\n";
}
```

```
void insert(int n, int pos)
{
    if (start == NULL)
    {
        cout<<"List not exist, can't insert value\n";
        return;
    }
    if(pos == 1)
    {
        cout<<"Invalid Position\n";
        return;
    }
    cur = start;
    for(int i = 1; i<= (pos-2); i++)
    {
        cur = cur->link;
        if (cur == start)
        {
            cout<<"Invalid Position\n";
        }
    }
    temp = new node;
    temp->data = n;
    temp->link = cur->link;
    cur->link = temp;
    cout<<n<<" inserted in list\n";
}
```



```
void search(int e)
{
    if (start == NULL)
    {
        cout<<"List not exist, can't search value\n";
        return;
    }
    cur = start;
    do
    {
        if (cur->data == e)
        {
            cout<<e<<" exist in list\n";
            return;
        }
        cur = cur->link;
    }
    while (cur != start);
    cout<<e<<" not exist in list\n";
}
```

```
void del(int value)
{
    if (start == NULL)
    {
        cout<<"List not exist, can't delete value\n";
        return;
    }
    cur = temp = start;
    do
    {
        if (cur->data == value)
        {
            cout << "Value found and Delete\n";
            if (cur == start)
            {
                start = cur->link;
            }
            temp->link = cur->link;
            delete cur;
            return;
        }
        temp = cur;
        cur = cur->link;
    }
    while (cur != start);
    if (cur == start)
        cout << "Value not Exist in list\n";
}
```

```
void print()
{
    if (start == NULL)
    {
        cout<<"List not exist, can't display value\n";
        return;
    }
    cur = start;
    if (cur == NULL)
        cout<<"List is Empty";
    cout << "\nData in link list\n\n";
    while (cur->link != start)
    {
        cout << cur->data << endl;
        cur = cur->link;
    }
    cout << cur->data << endl;
}
};
```



```

int main()
{
    cirlist obj;
    int ch, val, pos;
    cout << "1)Insert Value\n";
    cout << "2)Insert at Specfic Position\n";
    cout << "3)Search an Element\n";
    cout << "4)Delete Specfic Value\n";
    cout << "5)Display\n";
    cout << "6)Exit\n\n\n";
    do
    {
        cout << "Enter choice : ";
        cin >> ch;
        switch (ch)
        {
            case 1 :
                cout << "Input for insertion: " << endl;
                cin >> val;
                obj.append(val);
                break;
            case 2 :
                cout << "Input for insertion at Specfic Position: " << endl;
                cout << "Note: The value will inserted after 1st Position\n";
                cout<<"Value: ";
                cin >> val;
                cout<<"Position: ";
                cin >> pos;
                obj.insert(val,pos);
                break;
            case 3 :
                cout << "Enter a value you want to search in a list: " << endl;
                cout<<"Note: Function search the first node with value from the list\n";
                cout<<"Value: ";
                cin >> val;
                obj.search(val);
                break;
            case 4 :
                cout<<"Enter a value you want to Delete from a list.\n";
                cout<<"Note: Function deletes the first node with value from the list\n";
                cout<<"Value: ";
                cin>>pos;
                obj.del(pos);
                break;
            case 5 :
                obj.print();
                break;
            case 6 :
                cout << "Exit\n";
                break;
            default :
                cout << "Incorrect Option!\n";
        }
    }
    while (ch != 6);
    return 0;
}

```