



DOUBLE LINKED LIST

By DJVprogrammers

Task 1

//Write a program to create a new double linked list & store values into nodes

Code

```
#include<iostream>

using namespace std;

struct linklist
{
    int data;

    linklist*pre,*next;

    linklist()
    {
        pre=next=NULL;
    }
};

class nodes
{
    linklist*start,*cur,*temp;

    public:

    nodes()
    {
        start=NULL;
    }
}
```

```
void create_nodes(int n)
{
    if(start==NULL)
    {
        start=new linklist;
        start->pre=NULL;
        start->next=NULL;
        start->data=n;
    }
    else
    {
        cur=start;
        while(cur->next!=NULL)
        {
            cur=cur->next;
        }
        temp=new linklist;
        temp->pre=cur;
        temp->next=NULL;
        cur->next=temp;
        temp->data=n;
    }
}
```

```
        }

    }

    void print()

    {

        cur=start;

        int c=1;

        cout<<"___NODES HAS BEEN CREATED___"<<endl;

        while(cur->next!=NULL)

        {

            cout<<"data "<<c<<" = "<<cur->data<<endl;

            c++;

            cur=cur->next;

        }

        cout<<"data "<<c<<" = "<<cur->data<<endl;

    }

};

int main()

{

    nodes l1;

    int c,n;

    cout<<"_____NO. of NODES_____ "<<endl;
```

```
cout<<"How many nodes you want to create"<<endl;

cin>>n;

cout<<"_____CREATE NODES_____"<<endl;

    for(int i=1;i<=n;i++)

    {

        cout<<"enter data "<<i<<"\t ";

        cin>>c;

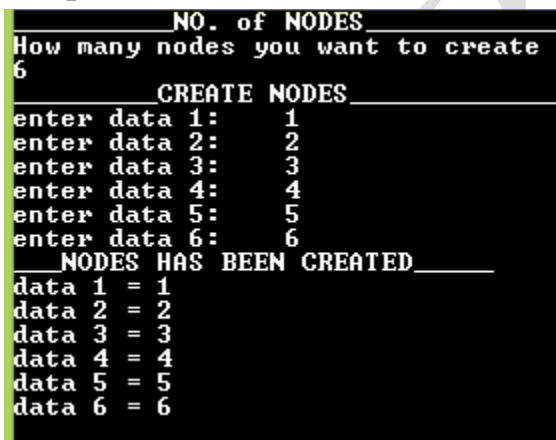
        l1.create_nodes(c);

    }

    l1.print();

}
```

Output:



```
_____NO. of NODES_____
How many nodes you want to create
6
_____CREATE NODES_____
enter data 1:    1
enter data 2:    2
enter data 3:    3
enter data 4:    4
enter data 5:    5
enter data 6:    6
_____NODES HAS BEEN CREATED_____
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 5
data 6 = 6
```

Task 2

Write a program to insert data at any place in double linklist

Code:

```
#include<iostream>
```

```
using namespace std;
```

```
struct linklist
```

```
{
```

```
    int data;
```

```
    linklist*pre,*next;
```

```
    linklist()
```

```
    {
```

```
        pre=next=NULL;
```

```
    }
```

```
};
```

```
class nodes
```

```
{
```

```
    linklist*start,*cur,*temp;
```

```
    public:
```

```
        nodes()
```

```
        {
```

```
            start=NULL;
```

```
        }
```

```
        void create_nodes(int n)
```

```
        {
```

```
    if(start==NULL)

    {

        start=new linklist;

        start->pre=NULL;

        start->next=NULL;

        start->data=n;

    }

    else

    {

        cur=start;

        while(cur->next!=NULL)

        {

            cur=cur->next;

        }

        temp=new linklist;

        temp->pre=cur;

        temp->next=NULL;

        cur->next=temp;

        temp->data=n;

    }

}
```

```
insertion(int d,int n)
{
    int i;

    cur=start;

    //insertion at start...

    if(d==1)
    {
        temp=new linklist;
        temp->next=start;
        temp->pre=NULL;
        start=temp;
        temp->next->pre=temp;
        temp->data=n;
    }
    else
    {
        //insertion at specific position and end...
        for(i=1;i<d-1;i++)
        {
            cur=cur->next;
        }
    }
}
```

```
temp=new linklist;

temp->pre=cur;

temp->next=cur->next;

cur->next=temp;

temp->data=n;

temp->next->pre=temp;

}

}

void print()

{

    cur=start;

    int c=1;

    cout<<"_____DATA_____"<<endl;

    while(cur->next!=NULL)

    {

        cout<<"data "<<c<<" = "<<cur->data<<endl;

        c++;

        cur=cur->next;

    }

    cout<<"data "<<c<<" = "<<cur->data<<endl;
```

```
    }

};

int main()
{
    nodes l1;

    int c1,N;

    cout<<"_____NO. of NODES_____"<<endl;

    cout<<"How many nodes you want to create"<<endl;

    cin>>N;

    cout<<"_____CREATE NODES_____"<<endl;

    for(int i=1;i<=N;i++)
    {

        cout<<"ENTER data "<<i<<":\t";

        cin>>c1;

        cout<<endl;

        l1.create_nodes(c1);

    }

    cout<<"_____OLD data_____"<<endl;

    l1.print();
}
```

```
cout<<"_____INSERT data_____"<<endl;

int d,n;

cout<<"enter position where you want to insert data"<<endl;

cin>>d;

cout<<"what data you want to insert"<<endl;

cin>>n;

l1.insertion(d,n);

cout<<"__list with inserted data_____"<<endl;

l1.print();

}
```

Output:

```
NO. of NODES
How many nodes you want to create
6
CREATE NODES
ENTER data 1: 1
ENTER data 2: 2
ENTER data 3: 3
ENTER data 4: 4
ENTER data 5: 5
ENTER data 6: 6
OLD data
DATA
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 5
data 6 = 6
INSERT data
enter position where you want to insert data
1
what data you want to insert
11
list with inserted data
DATA
data 1 = 11
data 2 = 1
data 3 = 2
data 4 = 3
data 5 = 4
data 6 = 5
data 7 = 6
```

Task 3

Create a double linklist and create a function that can delete data from it.

Code:

```
#include<iostream>

using namespace std;

struct linklist

{

    int data;
```

```
linklist*pre,*next;

linklist()

{

    pre=next=NULL;

}

};

class nodes

{

    linklist*start,*cur,*temp;

    public:

        nodes()

        {

            start=NULL;

        }

        void create_nodes(int n)

        {

            if(start==NULL)

            {

                start=new linklist;

                start->pre=NULL;

                start->next=NULL;
```

```
        start->data=n;

    }

    else

    {

        cur=start;

        while(cur->next!=NULL)

        {

            cur=cur->next;

        }

        temp=new linklist;

        temp->pre=cur;

        temp->next=NULL;

        cur->next=temp;

        temp->data=n;

    }

}

insertion(int n)

{

    int i;

    cur=start;

    if(start->data==n)
```

```
{

    start->next->pre=NULL;

    start=start->next;

    delete cur;

}

else

{

while(cur->data!=n)

{

    temp=cur;

    cur=cur->next;

}

if(cur->next==NULL)

{

    temp->next=NULL;

    delete cur;

}

else

{

    cur->next->pre=temp;
```

```
temp->next=cur->next;

delete cur;

}

}

}

void print()
{
    cur=start;
    int c=1;
    while(cur->next!=NULL)
    {
        cout<<"data "<<c<<" = "<<cur->data<<endl;
        c++;
        cur=cur->next;
    }
    cout<<"data "<<c<<" = "<<cur->data<<endl;

}

};
```

```
int main()
{
    nodes l1;

    int c1;

    for(int i=1;i<=10;i++)
    {
        cout<<"ENTER data "<<i<<":\t";

cin>>c1;

        cout<<endl;

        l1.create_nodes(c1);
    }

    cout<<"-----OLD data-----"<<endl;

        l1.print();

    cout<<"-----DELETE data-----"<<endl;

    int n;

    cout<<"what value of data you want to delete"<<endl;

    cin>>n;

    l1.insertion(n);

    cout<<"-----list with DELETED data-----"<<endl;

    l1.print();
```

```
}
```

Output:

```
ENTER data 1: 1
ENTER data 2: 2
ENTER data 3: 3
ENTER data 4: 4
ENTER data 5: 5
ENTER data 6: 6
ENTER data 7: 7
ENTER data 8: 8
ENTER data 9: 9
ENTER data 10: 0

-----OLD data-----
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 5
data 6 = 6
data 7 = 7
data 8 = 8
data 9 = 9
data 10 = 0

-----DELETE data-----
what value of data you want to delete
5
-----list with DELETED data-----
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 6
data 6 = 7
data 7 = 8
data 8 = 9
data 9 = 0

Process exited after 5.974 seconds with return value 0
Press any key to continue . . .
```

Task 4

Create a double linklist and search data from it...and modify the searched data..

Code:

```
#include<iostream>
```

```
using namespace std;

struct linklist
{
    int data;

    linklist*pre,*next;

    linklist()
    {
        pre=next=NULL;
    }
};

class nodes
{
    linklist*start,*cur,*temp;

    public:

        nodes()
        {
            start=NULL;
        }

        void create_nodes(int n)
        {
            if(start==NULL)
```

```
{  
  
    start=new linklist;  
  
    start->pre=NULL;  
  
    start->next=NULL;  
  
    start->data=n;  
  
}  
  
else  
  
{  
  
    cur=start;  
  
    while(cur->next!=NULL)  
  
    {  
  
        cur=cur->next;  
  
    }  
  
    temp=new linklist;  
  
    temp->pre=cur;  
  
    temp->next=NULL;  
  
    cur->next=temp;  
  
    temp->data=n;  
  
}  
  
}  
  
searching(int s)
```

```
{  
  
    cur=start;  
  
    while(cur->data!=s)  
  
    {  
  
        cur=cur->next;  
  
    }  
  
    cout<<"searched data.. now modify it"<<endl;  
  
    cout<<"enter new value"<<endl;  
  
    cin>>cur->data;  
  
}  
  
void print()  
  
{  
  
    cur=start;  
  
    int c=1;  
  
    while(cur->next!=NULL)  
    {  
  
        cout<<"data "<<c<<" = "<<cur->data<<endl;  
  
        c++;  
  
        cur=cur->next;  
  
    }  
  
    cout<<"data "<<c<<" = "<<cur->data<<endl;
```

```
    }

};

int main()
{
    nodes l1;

    int c1;

    for(int i=1;i<=10;i++)
    {
        cout<<"ENTER data "<<i<<":\t";

        cin>>c1;

        cout<<endl;

        l1.create_nodes(c1);
    }

    cout<<"-----OLD data-----"<<endl;

    l1.print();

    cout<<"-----Searching data-----"<<endl;

    int d;

    cout<<"enter data you want to search from list"<<endl;

    cin>>d;
```

```
l1.searching(d);

cout<<"-----list with modified data-----"<<endl;

l1.print();

}
```

Output:

```
ENTER data 1: 1
ENTER data 2: 2
ENTER data 3: 3
ENTER data 4: 4
ENTER data 5: 5
ENTER data 6: 6
ENTER data 7: 7
ENTER data 8: 8
ENTER data 9: 9
ENTER data 10: 0

-----OLD data-----
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 5
data 6 = 6
data 7 = 7
data 8 = 8
data 9 = 9
data 10 = 0
-----Searching data-----
enter data you want to search from list
4
searched data.. now modify it
enter new value
444
-----list with modified data-----
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 444
data 5 = 5
data 6 = 6
data 7 = 7
data 8 = 8
data 9 = 9
data 10 = 0

-----
Process exited after 11.41 seconds with return value 0
Press any key to continue . . .
```

Task 5

Traverse data in double linklist.

Code:

```
#include<iostream>
```

```
using namespace std;
```

```
struct linklist
```

```
{
```

```
    int data;
```

```
    linklist*pre,*next;
```

```
    linklist()
```

```
    {
```

```
        pre=next=NULL;
```

```
    }
```

```
};
```

```
class nodes
```

```
{
```

```
    linklist*start,*cur,*temp;
```

```
    public:
```

```
        nodes()
```

```
        {
```

```
            start=NULL;
```

```
        }
```

```
void create_nodes(int n)
{
    if(start==NULL)
    {
        start=new linklist;
        start->pre=NULL;
        start->next=NULL;
        start->data=n;
    }
    else
    {
        cur=start;
        while(cur->next!=NULL)
        {
            cur=cur->next;
        }
        temp=new linklist;
        temp->pre=cur;
        temp->next=NULL;
        cur->next=temp;
        temp->data=n;
    }
}
```

```
    }

}

void traversing()

{

    cur=start;

    while(cur->next!=NULL)

    {

        cur=cur->next;

    }

    int i;

    for( i=10;cur->pre!=NULL;i--)

    {

        cout<<"Data "<<i<<" = "<<cur->data<<endl;

        cur=cur->pre;

    }

    cout<<"Data "<<i<<" = "<<cur->data<<endl;

}

void print()

{

    cur=start;

    int c=1;
```

```
        while(cur->next!=NULL)
        {
            cout<<"data "<<c<<" = "<<cur->data<<endl;

            c++;

            cur=cur->next;

        }

        cout<<"data "<<c<<" = "<<cur->data<<endl;

    }

};

int main()
{
    nodes l1;

    int c1;

    for(int i=1;i<=10;i++)
    {
        cout<<"ENTER data "<<i<<":\t";

        cin>>c1;

        cout<<endl;

        l1.create_nodes(c1);

    }
}
```

```
cout<<"-----OLD data-----"<<endl;

l1.print();

cout<<"-----reverse list data-----"<<endl;

l1.traversing();

}
```

Output:

```
ENTER data 1: 1
ENTER data 2: 2
ENTER data 3: 3
ENTER data 4: 4
ENTER data 5: 5
ENTER data 6: 6
ENTER data 7: 7
ENTER data 8: 8
ENTER data 9: 9
ENTER data 10: 0

-----OLD data-----
data 1 = 1
data 2 = 2
data 3 = 3
data 4 = 4
data 5 = 5
data 6 = 6
data 7 = 7
data 8 = 8
data 9 = 9
data 10 = 0
-----reverse list data-----
Data 10 = 0
Data 9 = 9
Data 8 = 8
Data 7 = 7
Data 6 = 6
Data 5 = 5
Data 4 = 4
Data 3 = 3
Data 2 = 2
Data 1 = 1
-----
```

[Program.codes.cpp@gmail.com.](mailto:Program.codes.cpp@gmail.com)

<https://programcodescpp.wixsite.com/programcodes>

Website:

<https://programcodescpp.wixsite.com/programcodes>

Email

Program.codes.cpp@gmail.com

Facebook Page

<https://web.facebook.com/DJVprogrammers>