

<b>Name</b>	Komal Tarachandani
<b>UID no.</b>	2021600065
<b>Experiment No.</b>	4

<b>AIM:</b>	Implement Creation, sorting and merging of double linked lists.
<b>Program 4</b>	
<b>PROBLEM STATEMENT :</b>	<p>Perform following Operations on doubly <a href="#">linked list</a></p> <p>1-sort the list. Take input data in random order and insert it at appropriate place in the list</p> <p>2-Remove the duplicates from the sorted doubly <a href="#">linked list</a></p> <p>3-merge two doubly linked lists in sorted manner</p>
<b>PROGRAM:</b>	<pre>#include &lt;iostream&gt;  using namespace std; class Node{     public:     int data;     Node *next;     Node *prev; }; Node *head,*cur,*cur2; void create_node(Node *newnode) {     if(head==NULL)     {         head=newnode;         cur=newnode;     }     else     {         if(newnode-&gt;data&lt;head-&gt;data)         {             newnode-&gt;next=head;</pre>

```

        head->prev=newnode;
        head=newnode;
    }
    else if(newnode->data>=cur->data)
    {
        cur->next=newnode;
        newnode->prev=cur;
        cur=cur->next;
    }
    else
    {
        cur2=head;
        while(cur2!=cur)
        {
            if(newnode->data<cur2->data)
                break;
            cur2=cur2->next;
        }
        cur2=cur2->prev;
        newnode->next=cur2->next;
        newnode->prev=cur2;
        cur2->next->prev=newnode;
        cur2->next=newnode;
    }

}
}

void display(Node *cur)
{
    while(cur!=NULL)
    {
        cout<<cur->data<<"-->";
        cur=cur->next;
    }
}

void duplicate(Node *start)
{

```

```

Node *cur; cur=start;
while(cur->data==start->data && cur->next!=NULL)
{
    cur=cur->next;
}
if(cur->next==NULL && cur->data==start->data)
start->next=NULL;
else
{ start->next=cur;
  cur->prev=start;}
}
int main()
{
    head=NULL;
    int n,n2;
    cout<<"Enter the number of nodes for linked list";
    cin>>n;
    Node *newnode;
    int item;
    //CREATING LINKED LIST

    for(int i=0;i<n;i++){
        cout<<"Enter data for the node: ";
        cin>>item;
        newnode=new Node();
        newnode->data=item;
        newnode->next=NULL;
        newnode->prev=NULL;
        create_node(newnode);
    }
    cout<<"LINKED LIST 1 before removing DUPLIACES\n";
    display(head);
    cur=head;
    Node *head1,*newnode2,*ptr1;
    //REMOVING DUPLIACES
    while(cur->next!=NULL)
    {
        if(cur->data==cur->next->data)

```

```

        duplicate(cur);
    else
        cur=cur->next;
    }
    cur=head;
    //COPYING LL1
    newnode2=new Node();
    newnode2->next=NULL;
    newnode2->prev=NULL;
    newnode2->data=cur->data;
    head1=newnode2;
    ptr1=newnode2;
    while(cur->next!=NULL)
    {
        cur=cur->next;
        newnode2=new Node();
        newnode2->next=NULL;
        newnode2->prev=NULL;
        newnode2->data=cur->data;
        ptr1->next=newnode2;
        ptr1=ptr1->next;
    }
    cout<<"\n FINAL LINKED LIST1:\n";
    display(head1);

    //CREATING 2nd LINKED LIST
    head=NULL;
    cout<<"\nEnter number of nodes for second ll";
    cin>>n2;
    for(int i=0;i<n2;i++){
        cout<<"Enter data for the node: ";
        cin>>item;
        newnode=new Node();
        newnode->data=item;
        newnode->next=NULL;
        newnode->prev=NULL;
        create_node(newnode);
    }

```

```

cout<<"\nSECOND LIST before REMOVING DUPLIATES\n";
display(head);
cur=head;
while(cur->next!=NULL)
{
    if(cur->data==cur->next->data)
        duplicate(cur);
    else
        cur=cur->next;
}
cout<<"\nSECOND LINKED LIST\n";
display(head);
cur->next=head1;
cur=head;
for(int i=1;i<n+n2;i++)
{
    cur=head;
    while(cur->next!=NULL)
    {
        if(cur->data>cur->next->data)
        {
            int temp=cur->data;
            cur->data=cur->next->data;
            cur->next->data=temp;
        }
        cur=cur->next;
    }
}
cout<<"\nSORTED MERGED LIST\n";
display(head);

return 0;
}

```

**RESULT:**

```
input
Enter the number of nodes for linked list5
Enter data for the node: 2
Enter data for the node: 4
Enter data for the node: 4
Enter data for the node: 2
Enter data for the node: 1
LINKED LIST 1 before removing DUPLIATES
1-->2-->2-->4-->4-->
  FINAL LINKED LIST1:
1-->2-->4-->
Enter number of nodes for second list5
Enter data for the node: 3
Enter data for the node: 5
Enter data for the node: 6
Enter data for the node: 5
Enter data for the node: 3
SECOND LIST before REMOVING DUPLIATES
3-->3-->5-->5-->6-->
SECOND LINKED LIST
3-->5-->6-->
SORTED MERGED LIST
1-->2-->3-->4-->5-->6-->

...Program finished with exit code 0
Press ENTER to exit console.
```

**CONCLUSION:**

Hence I was able to learn the proper implementation and application of doubly linked lists.