

Assignment 7

Class - SEIV

Roll NO - 21430

Batch - F4

DOS - 24/11/2020

Problem statement -:

Write C++ program for sorting binary number using doubly linked list. Write functions

- a) To compute 1's and 2's complement.
- b) Add to binary numbers.

Learning objectives-:

1. To learn to write simple c++ program and execute it.
2. To implement Doubly Circular Linked list in c++.

Learning Outcomes-:

Will be able to implement doubly circular linked list in c++.

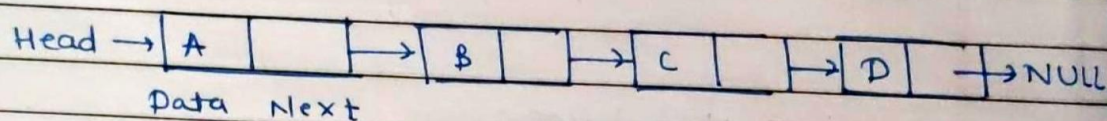
S/W and H/W Requirement.

1. Open Source c++ tools.
2. Windows 10 64 bit
3. Open source IDE eg. Eclipse

Theory :-

* Linked List :-

A linked list is a linear data structure in which the elements are not stored at contiguous memory location. The elements in a linked list are linked using pointers as shown in the below ~~im~~. The first node is always stored in head pointer.



There are generally four types of linked list.

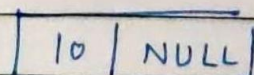
1. Singly Linked List.
2. Doubly Linked List.
3. Circular Linked List.
4. Doubly circular Linked List.

1. Singly Linked List :-

In Singly linked list there are 2 data fields, i.e. data and next pointer.

A single node has always NULL in its next pointer field.

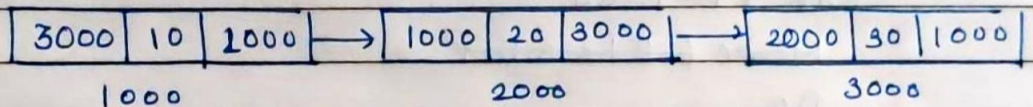
Structure of Node for SLL.



data Next.

* Doubly Circular Linked List :-

The structure of Node of DCLL is same as that of DLL, the only difference betⁿ DLL and DCLL is that the previous pointer of head node points to the last node of the list and next pointer of last node points to 1st node of the list. There is never NULL in any node of DCLL.



* Algorithm / Pseudocode :-

* ADT for class Node

Class Node

int data // data field

Node *next // Pointer to next node

Node *prev // Pointer to prev Node

* ADT for class DCLL :-

class DCLL

Node *head // head pointer

void show() // show data

int count() // Gives count

Void addatbeg() // Add element at beginning
 Void addatend() // Add element at end
 DCLL ocomp() // Return 1's complement LL.
 DCLL add() // Add 2 bin. numbers

* Psuedocode for Display

```

Void display()
Node *P = head
While (P != NULL)
Do
    cout << P->data
    P = P->next
While (P != head)
END
  
```

* Complexity

i. Time complexity

create() - $O(1)$
 ocomp() - $O(1)$
 tcomp() - $O(1)$
 add() - $O(1)$

ii. Space Complexity

create() - $O(1)$
 ocomp() - $O(1)$
 tcomp() - $O(1)$
 add() - $O(1)$

* Test Cases -:

No.	Description	Input	Expected o/p	Actual o/p	Result.
1.	1. 1's comp 2. 2's comp 3. Add	ch = 1 Bin-0001	1's comp = 1110	1's comp = 1110	pass
2.	1. 1's comp 2. 2's comp 3. Add	ch = 2 Bin-0001	2's comp = 1111	2's comp = 1111	pass.

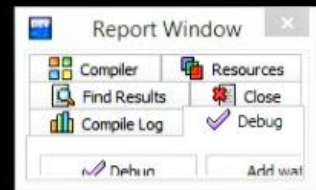
Conclusion -:

We learnt to implement the concept of Doubly Circular Linked List.


```

1  #include <iostream>
2  using namespace std;
3  class Node
4  {
5      public:
6          int data;
7          Node *prev;
8          Node *next;
9          Node()
10         {
11             data=0;
12             prev=next=NULL;
13         }
14         Node(int a)
15         {
16             data=a;
17             prev=next=NULL;
18         }
19     };
20     class DLL
21     {
22     public:
23         Node *head;
24         DLL()
25         {
26             head=NULL;
27             Node *p;
28             p=head;
29         }
30     }
31     void addatbeg(int);
32     void addatend(int);
33     void search();

```

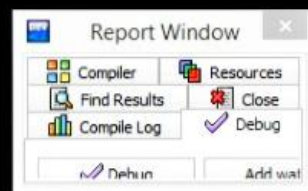


Activate Windows
Go to PC settings to activate Windows.

```

37     DLL tcomp();
38     void rshow();
39     DLL add(DLL,DLL);
40 };
41 void DLL::rshow()
42 {
43     Node *p;
44     p=head;
45     do
46     {
47         p=p->next;
48     }while(p->next!=NULL);
49     while(p!=NULL)
50     {
51         cout<<p->data<<" ";
52         p=p->prev;
53     }
54 }
55 void DLL::create()
56 {
57     int i=0;
58     int x;
59     Node *p;
60     cout<<"\nEnter given Dcimal Number ";
61     int num;
62     cin>>num;
63     if(num==0)
64     {
65         addatbeg(0);
66         return ;
67     }
68     while(num!=0)
69     {

```



Activate Windows
Go to PC settings to activate Windows.

```

69 {
70     x=num%2;
71     addatbeg(x);
72     num=num/2;
73 }
74 if(count()<8&&count()>4)
75 {
76     for(i=count();i<8;i++)
77     {
78         addatbeg(0);
79     }
80 }
81 else if(count()<4)
82 {
83     for(i=count();i<4;i++)
84     {
85         addatbeg(0);
86     }
87 }
88 }
89 }
90 DLL DLL::ocomp()
91 {
92     Node *p;
93     p=head;
94     DLL q;
95     do
96     {
97         if(p->data==0)
98         {
99             q.addatend(1);
100         }
101     }

```

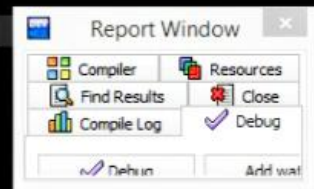


Activate Windows
Go to PC settings to activate Windows.


```

104         }
105         p=p->next;
106     }while(p!=NULL);
107     return q;
108 }
109 void DLL::addatend(int x)
110 {
111     Node* q=NULL;
112     q=new Node(x);
113     if(head==NULL)
114     {
115         head=q;
116     }
117     else
118     {
119         Node*p=head;
120         while(p->next!=NULL)
121         {
122             p=p->next;
123         }
124         p->next=q;
125         q->prev=p;
126     }
127 }
128 }
129 DLL DLL::tcomp()
130 {
131     DLL q,p;
132     p=ocomp();
133     q.addatbeg(1);
134     return add(q,p);
135 }
136

```

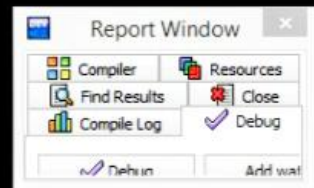


Activate Windows
Go to PC settings to activate Windows.

```

138 {
139     int carry=0;
140     Node *p,*q;
141     p=a.head;
142     q=b.head;
143     DLL s;
144     while(p->next!=NULL)
145     {
146         p=p->next;
147     }
148     while(q->next!=NULL)
149     {
150         q=q->next;
151     }
152     while(p!=NULL || q!=NULL)
153     {
154         if(p!=NULL && q!=NULL)
155         {
156             if(p->data==1 && q->data==1)
157             {
158                 if(carry==0)
159                 {
160                     carry=1;
161                     s.addatbeg(0);
162                 }
163                 else
164                 {
165                     carry=1;
166                     s.addatbeg(1);
167                 }
168             }
169             else if((p->data==1 && q->data==0) || (p->data==0 && q->data==1))
170             {

```

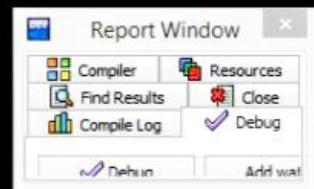


Activate Windows
Go to PC settings to activate Windows.


```

173         carry=1;
174         s.addatbeg(0);
175     }
176     else
177     {
178         carry=0;
179         s.addatbeg(1);
180     }
181 }
182 else
183 {
184     if(carry==1)
185     {
186         carry=0;
187         s.addatbeg(1);
188     }
189     else
190     {
191         s.addatbeg(0);
192     }
193 }
194 }
195 else if(p==NULL&&q!=NULL)
196 {
197     if(carry==1)
198     {
199         if(q->data==1)
200         {
201             s.addatbeg(0);
202             carry=1;
203         }
204         else
205         {

```

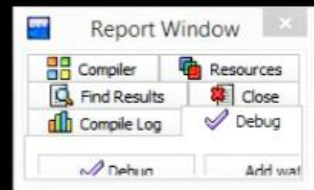


Activate Windows
Go to PC settings to activate Windows.

```

211 {
212     if(q->data==1)
213     {
214         s.addatbeg(1);
215         carry=0;
216     }
217     else
218     {
219
220         s.addatbeg(0);
221     }
222 }
223 }
224 else if(p!=NULL&&q==NULL)
225 {
226     if(carry==1)
227     {
228         if(p->data==1)
229         {
230             s.addatbeg(0);
231             carry=1;
232         }
233         else
234         {
235             carry=0;
236             s.addatbeg(1);
237         }
238     }
239     else
240     {
241         if(p->data==1)
242         {
243             s.addatbeg(1);

```

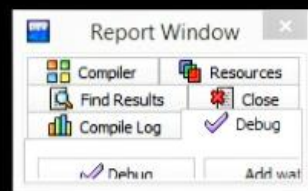


Activate Windows
Go to PC settings to activate Windows.


```

241         if(p->data==1)
242         {
243             s.addatbeg(1);
244             carry=0;
245         }
246         else
247         {
248
249             s.addatbeg(0);
250         }
251     }
252 }
253 if(p!=NULL)
254 {
255     p=p->prev;
256 }
257 if(q!=NULL)
258 {
259     q=q->prev;
260 }
261 }
262 return s;
263 }
264
265 int main()
266 {
267
268     int ch;
269     while(1)
270     {
271         cout<<"! ! ! Enter Choice ! ! !";
272         cout<<"\n1.One's Complement";
273         cout<<"\n2.Two's Complement";

```

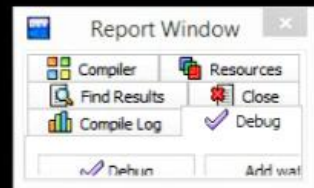


Activate Windows
Go to PC settings to activate Windows.

```

275 cout<<"\n0.Exit";
276 cout<<"\n\nCHOICE : ";
277 cin>>ch;
278 switch(ch)
279 {
280     case 1:
281     {
282         DLL D1,D2,D3,D5;
283         D1.create();
284         cout<<"\nThe Binary is : ";
285         D1.show();
286         D2=D1.ocomp();
287         cout<<"\n\nOne's Complement of given binary number is ";
288         D2.show();
289         cout<<"\n\n";
290         break;
291     }
292     case 2:
293     {
294         DLL D1,D2,D3,D5;
295         D1.create();
296         cout<<"\nThe Binary is : ";
297         D1.show();
298         D2=D1.tcomp();
299         cout<<"\n\nTwo's Complement of given binary number is ";
300         D2.show();
301         cout<<"\n\n";
302         break;
303     }
304     case 3:
305     {
306         DLL D1,D2,D3,D5;
307         D1.create();

```

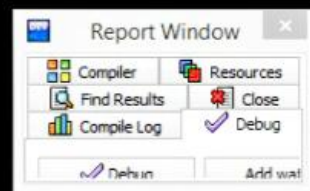


Activate Windows
Go to PC settings to activate Windows.


```

308         cout<<"\nThe Binary is : ";
309         D1.show();
310         cout<<"\n";
311         D5.create();
312         cout<<"\nThe Binary is : ";
313         D5.show();
314         D2=D2.add(D1,D5);
315         cout<<"\n\nAddition is : ";
316         D2.show();
317         cout<<"\n\n";
318         break;
319     }
320     case 0:
321     {
322         exit(1);
323     }
324     default:
325     {
326         cout<<"\n\nInvalid Choice \n\n";
327     }
328 }
329 }
330
331 return 0;
332 }
333 int DLL::count()
334 {
335     Node *p;
336     int cnt=0;
337     p=head;
338     while(p!=NULL)
339     {
340         cnt++;

```

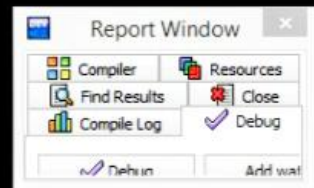


Activate Windows
Go to PC settings to activate Windows.

```

343     return cnt;
344 }
345 void DLL::show()
346 {
347     Node *p;
348     p=head;
349     if(head==NULL)
350     {
351         cout<<"\nEmpty list";
352         return;
353     }
354     while(p!=NULL)
355     {
356         cout<<p->data<<" ";
357         p=p->next;
358     }
359 }
360 }
361 void DLL::addatbeg(int x)
362 {
363     Node* q=NULL;
364     q=new Node(x);
365     if(head==NULL)
366     {
367         head=q;
368     }
369     else
370     {
371         q->next=head;
372         head->prev=q;
373         head=q;
374     }
375 }

```



Activate Windows
Go to PC settings to activate Windows.

! ! ! Enter Choice ! ! !
1.One's Complement
2.Two's Complement
3.Add to binary no
0.Exit

CHOICE : 1

Enter given Dcimal Number 5

The Binary is : 0 1 0 1

One's Complement of given binary number is 1 0 1 0

! ! ! Enter Choice ! ! !
1.One's Complement
2.Two's Complement
3.Add to binary no
0.Exit

CHOICE : 2

Enter given Dcimal Number 4

The Binary is : 0 1 0 0

Two's Complement of given binary number is 1 1 0 0

! ! ! Enter Choice ! ! !
1.One's Complement
2.Two's Complement
3.Add to binary no
0.Exit

CHOICE : 3

Enter given Dcimal Number 4

The Binary is : 0 1 0 0

Enter given Dcimal Number 5

The Binary is : 0 1 0 1

Addition is : 1 0 0 1

! ! ! Enter Choice ! ! !
1.One's Complement
2.Two's Complement
3.Add to binary no
0.Exit

CHOICE : 0

Process exited after 0.243 seconds with return value 1
Press any key to continue . . .