-		Classmate
		S.R. POOJA
		1BM19CS135
		LAB PROG-9
		#include < stdio.h>
	Was John	#include (stalib. h)
	l l	
-		Struct node
-		E int info;
-		Skuet node * llink',
1		Strict node * slink; 4,
-		typed struct node * NODE;
4		NODE get roole &
4		NoDE x'
4		2= (NODE) mallor (Sign of (skuck node));
1		if (x== NULL)
1		& pents ("men fall \n");
		ent (0) proses a classical thing and account
	(4) 40	3 setuen x; 3
		void freenode (NODE N)
		E fee (x); 3 (++) man = s; it = 1 + in) tale
	· (" Le?	NODE dinsert-front (int iten, NODE head)
		E NODE temp, cue;
		temp= get node();
		temp >inp = Ptem;
		cue= head-> & link;
		head-selink: tenp', & "(Inote books) 75 - books
П		tenp-) llenk = head; (c== smole) 16
П		temp -> elink = cur' to not set selo ") Has
I		cue > llink : tenp', (lange & labore)
-		setus head; S'(Instibusia) at almoss
		3 NODE donseit leas (int ston, NODE head)
		E NOSE temp, cus;
-		temp= getnodel);
-		lean into sitem'
-		temp > info = item'; cus = head > think; (+24)
-		head -> Unk: temp;
-		temp > slink = all head;
-		temp -> llork= cus;
		cur > stink = temp;
		aut-y state (10.7)

	S.R. POOJA classmate
	1BM19C8135
	roters head; 3
	NODE ddelete fent (NODE head)
	{ NODE are, next;
	if (head > glink == head)
	Epents (adg emptylno);
	setus head; 3
	cur = head > slink;
	next = cue > rbnh,
	head > rlink = next;
	next > llonk = head;
	parts (4the node deteted is 1.d4, all-> info);
	Jeenode (cue) 3000 mil day 200 Algo Assess 3000
	3 NODE ddelete eeg (NODE head)
	3 NODE ddelete eag (NODE head)
	& NODE cue, preu',
	if (head & struz= head)
	E peint (udg empty \n");
	esturn head & Compile
	cu = head > Unk; island (africe 200 = note) for
	prev = cya -> Clink;
	head > Unk = perl;
	pres selinke head; ("Many ton my) ling?
	party l'the node deleted
	NODE inset-toftpos (int item, Nove head)
	E NODE temp, cue, peu
	ij (head-telenk== head)
	[prot (u lot enpty) nu)
	return head; 3
	cur = head > slink;
	whole (cue) = head)
	f if (inter== = cue > info) break)
	cue = cue -> slinki,
	· S
-	if (cue == head) and si) and 3000
	t

		ar alassmate
	BATT	S. K. Pao JA
	1	1 BM 19 CS135
		11.11) Charl conta
		pants (" key not found in,
-		Selven head, I
4		
-		pently (nenter bowards lift of 1.d2", (ten); temp= getnode ();
-		temp= getnode (),
-		scanf (ind", &temp > info),
H		peur) ylink: temp,
-		temp -> links pew,
		cue > llink = cue; / from shilr & longer
		setuen head; hord = stall <- trans
		3 con Caline run Bix is hittel about antil Itany
		NODE insect eight pos (int iten, node head)
		& NODE temp, cue, nest;
		if (head -) slink = head) and Jenes at John &
		2 pent ("list empty in");
		getuen head
		3 cus = head -> slink; ("a/plans (b") fing?
		while (cus!=head)
		E if (iton == cus > info) break;
		cue = cue > elerli; 3
		if (cul==head)
		E pant (" key not found n"); / hour sales
		getien head; bother estal Hara
		3 next-(cue => etinte) moto dis) endlat - torm + 0014
		print ("enter towards sight of Vide", item);
		temp = getnode(); (book = stanted book)
		scanf & ".1.d", & temp->info);
		cue > slink = temp,
		temp > llink = cue; dada = book = 200
1		next -> Clink = temp; (book = 1 sus) whate
1		temp -> slenk = next; dood () () ()
		Setur head;
1		I was a second of
-		NODE search (NODE head, int item)
1		{
		The state of the s

100	classmate
7-	S.R. POOJA
-	18M19CS (35
	cul= next; 3 (20) good 3004
	30 - 5021 30
	if (count==0) (hood == strike & bood)
	pent ("not found in");
	else i boat mile
	pently (upound at 1d positions and are deleted", count);
	setus head; 3
	4 (drif- sus sanotis) K &
	Sold display (NODF head)
	NODE tean' 'troope you for
	if (head-) slock: chead)
	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	print (uday en pty In"); ("allessammen de sore") Haza
	selin)
	and the sale and later
	print ('rantents of da \n");
	temp= to head -> stirtly not to be to to to to to to
	white (temp! = head) # 1000 1000 1000 1000
	The Count
	pently (uv.d\n", temp->info) (books and chow)
	tenp= temp->elinh; 3
	proff ("In")'
	g int main()
	NODE head, lost; Closed of me) states.
	lead at 1 ():
	1 1 2 21 1 1 1
	1
	5° pant ("In!: insect fant In 2: insect eag In 3. delete front In 4: Delete sear In 5: insect left of key clement In6', insect
	Right 9 key element \n 7: search \n8: delete seperating
	(as) Banes

```
#include<stdio.h>
 1
 2
     #include<stdlib.h>
 3
     struct node
 4
     {
 5
        int info;
 6
        struct node *llink;
 7
        struct node *rlink;
 8
        }:
     typedef struct node *NODE;
 9
     NODE getnode()
10
11
     {
12
       NODE x;
13
       x=(NODE)malloc(sizeof(struct node));
       if(x==NULL)
14
15
         printf("mem full\n");
16
17
         exit(0);
18
         }
19
       return x;
20
       }
     void freenode(NODE x)
21
22
23
       free(x);
24
25
     NODE dinsert_front(int item, NODE head)
     {
26
     NODE temp, cur;
27
     temp=getnode();
28
29
     temp->info=item;
30
     cur=head->rlink;
31
     head->rlink=temp;
32
     temp->llink=head;
     temp->rlink=cur;
33
     cur->llink=temp;
34
35
     return head;
36
     }
     NODE dinsert_rear(int item, NODE head)
37
38
     {
39
     NODE temp, cur;
     temp=getnode();
40
```

```
40
     temp=getnode();
     temp->info=item;
41
42
     cur=head->llink:
43
     head->llink=temp;
44
     temp->rlink=head;
     temp->llink=cur;
45
46
     cur->rlink=temp:
47
     return head;
48
     }
     NODE ddelete_front(NODE head)
49
50
     {
51
     NODE cur, next;
52
     if(head->rlink==head)
53
     {
     printf("dq empty\n");
54
     return head;
55
56
57
     cur=head->rlink;
58
     next=cur->rlink;
59
     head->rlink=next;
     next->llink=head;
60
     printf("the node deleted is %d",cur->info);
61
62
     freenode(cur):
     return head;
63
64
65
     NODE ddelete rear(NODE head)
66
     {
67
     NODE cur, prev;
     if(head->rlink==head)
68
69
     1
70
     printf("dq empty\n");
     return head;
71
72
     }
73
     cur=head->llink;
74
     prev=cur->llink;
     head->llink=prev;
75
     prev->rlink=head;
76
     printf("the node deleted is %d",cur->info);
77
78
     freenode(cur):
     return head:
79
```

```
return head;
79
80
      }
81
     NODE insert_leftpos(int item, NODE head)
82
83
      {
     NODE temp, cur, prev;
84
      if(head->rlink==head)
85
86
      {
     printf("list empty\n");
87
      return head;
88
      }
89
90
      cur=head->rlink:
     while(cur!=head)
91
92
      {
93
      if(item==cur->info)break;
94
      cur=cur->rlink;
95
      }
      if(cur==head)
96
97
      {
       printf("key not found\n");
98
99
       return head;
       }
100
101
       prev=cur->llink;
       printf("enter towards left of %d=",item);
102
103
       temp=getnode();
       scanf("%d",&temp->info);
104
105
       prev->rlink=temp;
106
       temp->llink=prev:
107
       cur->llink=temp;
       temp->rlink=cur;
108
109
       return head;
110
      }
111
     NODE insert_rightpos(int item, NODE head)
112
113
      {
     NODE temp, cur, next;
114
115
      if(head->rlink==head)
      {
116
117
      printf("list empty\n");
      return head;
118
```

```
main.c
      recurs means
119
      }
120 cur=head->rlink;
      while(cur!=head)
121
122
      {
123
     if(item==cur->info)break;
124
      cur=cur->rlink;
125
      }
126
      if(cur==head)
127
       printf("key not found\n");
128
       return head:
129
       }
130
131
       next=cur->rlink;
       printf("enter towards right of %d=",item);
132
133
       temp=qetnode();
       scanf("%d",&temp->info);
134
135
       cur->rlink=temp;
136
       temp->llink=cur;
       next->llink=temp;
137
138
       temp->rlink=next;
      return head:
139
140
141
      NODE search(NODE head, int item)
      {
142
      NODE temp, cur;
143
144
       int flag=0;
      if(head->rlink==head)
145
146
147
      printf("list empty\n");
148
     return head;
149
      }
150
      cur=head->rlink:
151
      while(cur!=head)
152
153
      if(item==cur->info)
154
      1
155
          flag=1;
156
          break;
157
      }
158
      cur=cur->rlink;
```

```
rai -cai ->i raik'
}
 if(cur==head)
printf("search unsuccessfull\n");
if(flag==1)
printf("search successfull\n");
NODE delete_all_key(int item, NODE head)
{
   NODE prev, cur, next;
    int count;
    if(head->rlink==head)
        printf("list empty\n");
        return head;
    count=0;
    cur=head->rlink;
    while(cur!=head)
    {
        if(item!=cur->info)
        cur=cur->rlink;
        else
        {
            count++;
            prev=cur->llink;
            next=cur->rlink;
            prev->rlink=next;
            next->llink=prev;
            freenode(cur);
            cur=next;
    if(count==0)
    printf("not found\n");
    else{
    printf("found at %d positions and are deleted", count);
      return head;
    }
}
```

```
198
      void display(NODE head)
199
200
      {
201
      NODE temp;
      if(head->rlink==head)
202
203
      1
      printf("dq empty\n");
204
205
      return;
206
207
      printf("contents of dq\n");
208
      temp=head->rlink;
      while(temp!=head)
209
210
      {
211
      printf("%d\n", temp->info);
212
      temp=temp->rlink;
213
      }
      printf("\n");
214
215
216
      int main()
217
      1
218
      NODE head, last;
219
      int item, choice;
220
      head=getnode();
221
      head->rlink=head;
222
      head->llink=head:
223
224
      for(;;)
225
226
        printf("\n1:insert front\n2:insert rear\n3:delete front\n4:delete
        rear\n5:insert left of key element\n6:insert right of key
        element\n7:search\n8:delete repeating occurances\n9:display\n10:exit\n");
        printf("enter the choice\n");
227
228
        scanf("%d", &choice);
        switch(choice)
229
230
        {
          case 1: printf("enter the item at front end\n");
231
232
            scanf("%d",&item);
233
            last=dinsert_front(item, head);
234
            break:
235
          case 2: printf("enter the item at rear end\n");
```

```
case 1: printf("enter the item at front end\n");
  scanf("%d",&item);
  last=dinsert_front(item, head);
  break;
case 2: printf("enter the item at rear end\n");
  scanf("%d",&item);
  last=dinsert_rear(item, head);
  break;
case 3:last=ddelete_front(head);
case 4: last=ddelete_rear(head);
  break;
case 5:
     printf("enter the key element\n");
     scanf("%d",&item);
     last=insert_leftpos(item, head);
     break:
 case 6:
     printf("enter the key element\n");
     scanf("%d",&item);
     last=insert_rightpos(item, head);
     break:
case 7:
        printf("enter the search element\n");
        scanf("%d",&item);
        search(head, item);
        break:
case 8: printf("enter element to be deleted\n");
        scanf("%d",&item);
        last=delete_all_key(item, head);
case 9: display(head);
  break:
default:exit(0);
}
```

```
enter the item at front end
                                                                             Q
12
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7:search
8:delete repeating occurances
9:display
10:exit
enter the choice
enter the item at rear end
21
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7:search
8:delete repeating occurances
9:display
10:exit
enter the choice
the node deleted is 12
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7:search
8:delete repeating occurances
9:display
10:exit
enter the choice
```

```
the node deleted is 21
                                                                             Q
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7: search
8:delete repeating occurances
9:display
10:exit
enter the choice
5
enter the key element
list empty
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7:search
8:delete repeating occurances
9:display
10:exit
enter the choice
dq empty
1:insert front
2:insert rear
3:delete front
4:delete rear
5:insert left of key element
6:insert right of key element
7:search
8:delete repeating occurances
9:display
10:exit
enter the choice
```