```
LAB 9:-
Program:-
#include<stdio.h>
#include<stdlib.h>
struct node
{
       int info;
       struct node *Ilink;
       struct node *rlink;
       };
typedef struct node *NODE;
NODE getnode()
{
      NODE x;
      x=(NODE)malloc(sizeof(struct node));
      if(x==NULL)
      {
            printf("mem full\n");
            exit(0);
            }
      return x;
      }
void freenode(NODE x)
{
```

```
free(x);
}
NODE dinsert_front(int item, NODE head)
{
NODE temp,cur;
temp=getnode();
temp->info=item;
cur=head->rlink;
head->rlink=temp;
temp->llink=head;
temp->rlink=cur;
cur->llink=temp;
return head;
}
NODE dinsert_rear(int item, NODE head)
{
NODE temp, cur;
temp=getnode();
temp->info=item;
cur=head->llink;
head->llink=temp;
temp->rlink=head;
temp->llink=cur;
cur->rlink=temp;
return head;
```

```
}
NODE ddelete_front(NODE head)
{
NODE cur,next;
if(head->rlink==head)
printf("dq empty\n");
return head;
}
cur=head->rlink;
next=cur->rlink;
head->rlink=next;
next->llink=head;
printf("the node deleted is %d",cur->info);
freenode(cur);
return head;
}
NODE ddelete_rear(NODE head)
{
NODE cur, prev;
if(head->rlink==head)
{
printf("dq empty\n");
return head;
}
```

```
cur=head->llink;
prev=cur->llink;
head->llink=prev;
prev->rlink=head;
printf("the node deleted is %d",cur->info);
freenode(cur);
return head;
}
NODE insert_leftpos(int item,NODE head)
{
NODE temp,cur,prev;
if(head->rlink==head)
{
printf("list empty\n");
return head;
}
cur=head->rlink;
while(cur!=head)
{
if(item==cur->info)break;
cur=cur->rlink;
}
if(cur==head)
{
 printf("key not found\n");
```

```
return head;
 }
 prev=cur->llink;
 printf("enter towards left of %d = ",item);
 temp=getnode();
 scanf("%d",&temp->info);
 prev->rlink=temp;
 temp->llink=prev;
 cur->llink=temp;
 temp->rlink=cur;
 return head;
}
NODE delete_specified_value(int item,NODE head)
{
NODE prev,cur,next;
int count;
   if(head->rlink==head)
    {
     printf("List is empty");
     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("key not found");
  else
 printf("Key found at %d positions and are deleted\n", count);
return head;
}
void display(NODE head)
{
```

```
NODE temp;
if(head->rlink==head)
{
printf("dq empty\n");
return;
}
printf("contents of dq\n");
temp=head->rlink;
while(temp!=head)
{
printf("%d\n",temp->info);
temp=temp->rlink;
}
printf("\n");
}
void main()
{
NODE head, last;
int item, choice;
head=getnode();
head->rlink=head;
head->llink=head;
for(;;)
```

{

```
printf("\n 1:insert front\t 2:insert rear\t 3:delete front\t 4:delete rear\t 5:Insert left position\t
6:Delete specified value\n 7:display\t 8:exit\n");
      printf("enter the choice\n");
      scanf("%d",&choice);
      switch(choice)
      {
            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);
                  break;
            case 4: last=ddelete_rear(head);
                  break;
          case 5:printf("enter the key item\n");
             scanf("%d",&item);
             head=insert_leftpos(item,head);
             break;
         case 6:printf("enter the key item\n");
             scanf("%d",&item);
```

head=delete_specified_value(item,head);

```
break;

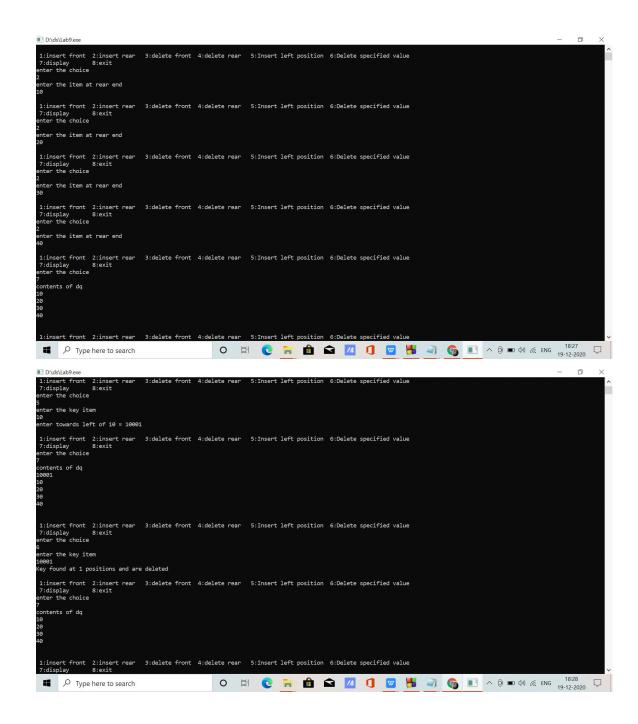
case 7: display(head);

break;

default:exit(0);

}
```

Output Screenshot:-



Written Pictures:-

```
Doubly linked list:
   # include cstdio.h7
   #include zetdlib.n>
   struct node
  int info;
   show unde *llinki
  struct node * rlink;
  typedet struct node * NOPE;
  NODE gernode () 4
  NODE X;
 x = (NODE) mallocl eize of (struct node));
 it (x == NOW) {

printt !" mum full in");
 exit (0); }
suturn x; }
void frumodi (NODE x)
 free (x); show a know a 4xxx
NODE dinsort-front (int item, NODO head) }
NODE temp, wit:
temp = getnodi ();
temp > info = iten;
un = hood -> runk;
 head > rlink = temp;
terap = link = head;
tenp > rlink = wi;
 we slink = temp;
outurn head;
```

NODE dinsuit- quartine item, NODE head); Nobe temp, wis temp = gernode(); temp > into = items cevi = head > 11inki heard -> elink = temp! temp > stink = head; temp > Llink = curi un > nint = temp; sections wood; Nobe delive-front (Nobe head) 4 Nobe cur, next; if (head > yenk = = head) } pf (" dq emply in"); return head; 4 un = head > nlink; howens next = cun > rlink; heard > slink = nexts next + llink = head; of ("the node deleted is it d", con sinh) prenode(wi); retroin head; NODE ddelete-near (NODE head) NODE wy, prun; of (maid > rlink = = heard) pf (" day empty in"); return mad)

```
um = head > Wink;
   pun = wr >llinki)
   head > UINK = pren;
   prun > slink = head;
    pt ("the node delited is 1.d", we sinto);
   foremode (wi);
   surum head;
   NODE insut-testpos(int item NODE mad)
   Nobe temp, wir, prens
   if (mad > rlink = head) &
   print ("list empty in");
   surum head;
  un = head > nlink;
  while (un! = head)
  print (" key not found in 1);
 sieturn headi j
 temp & link = prev; 1001) palgate block
 un > llink = timp;
temp > rink = wor ; I will to brown ) 47
 seturn head;
NODE delete-specified-value (int item, NODE had)
Nobe previous, nexts
int counts ( )
pt (head > rlink == head)
De [11 1 C 11].
```

```
NODE temp;
   it (figure == NULL)
   point (" list empty");
   for (temp = first; temp!= NULL; temp = temp > line)
   pountf(" =d in", temp >into);
  NODE cornat (NODE first, NODE suond)
  NODE wen;
  if (firest = = NULL)
  nutrun second:
  ?H swound = = NULL)
  neturn fort;
 us = first;
  while ( wir -> link! = NULL)
  wy = wy >link;
 cur > link = suond;
 neturn first?
 NODE reverse (NODE first)
 NODE witemp; I was some
 cur = NULL'
while ( first != NULL)
 temp = tioust;
forst = forst > 19nk;
temps link: wi;
cur = temp; (1000 mas) dela
neturn um.
```

while (un!=mad) } Pt (item! - wasinto) us = cun > rtinti else count ++; pren= un >llink1 Canin c run = +xun reserve + rink = next) nex+ allink = previ freemode (ws); (un = next) 47 (court = = 0) P& (" key not found"); else Pf (" key found @ r. of poston deleted", will return heard; } void display (NODE nead) NODE tempi Pf (neard > XVink == head) Pf ("dg empty mi) nehom; 1 min way and Pt (" contents of ob,") temp - head - runte) while (temp 1 = heard) Pt (". d'+inp+into); temp = temp -> stinks

void main biov NODE head, last; Port item, choice: head = getnode 17: head + slink = nead; head > llink = head; 4091(;j) PEC" IF 1+2 IR 1+3 DF 14 DR 15 MONTH TEXT pos 1+6. dette specified value 17: display it 8 exit in"); switch (choice); I case I: Pf l" enter the item (a front end in'); scan+ (" / d", fitem); Last = dinsult_front (item, head); break; case 2: perivot (" enter the item @ near end m"); (" , d", fitum)) 42 last = dinsort - mar (item, head); bruak: case 3: last = ddelete from (head); break) case 4: east-ddelik rear (head); break) cans: p+ (" key 'n'); s+ ("1.d", gitem"; ase 6: mad = delite spicified - value (item, man) becakes case 7: dis play (had; break; souse defaux: exitto);