```
Doubly linked list:
 # include < Stalio.h?
 #tinclude < comio.h>
 struct node
 int info:
  Struct node "llink's
 struct node *rlink;
typedel struct node *NODE;
NODE getnoold
  NODE X;
  x= (NODE) melloc (size of (struct node));
  if (x==NULL)
   printf ("remon full \n");
   exit (0);
 return x;
void freemode (NODE x)
  fee(x);
```

NOOE dinsert front (intitem, NODE head)	
NODE temp cur;	
temp=getnode();	
temp-) into = item	
cur = head > rlink;	
head > rlink = temp;	
Jemp- Mint = head;	
cur-> llink= denpo;	1500
temp-1 rlink= Red;	
return head;	1314
2	1
	2
NODE densert rear (int item, NODE head)	
S COMPANY TO THE COMMENT TO THE STATE OF THE	
NODE temp, cur;	£
temp=getnode();	y **
temp - into = item;	× .
cur = head > elink;	
head & llink = temp;	,
Jemps Rlink = head;	4
try-tlink= cur;	ć
we -> Elink = Amp;	
return headi	
To any could make the local state of their	100 J
	1
NODE delete front (NODE head)	2/4
NODE wer, next:	
if (head-) rlink == head)	0
t that bound to 1 " his made	
printf ("The DLL is empty \n");	
return head;	
3	

```
next = cur - rlink;
       head - rlink = next;
        next - Llink = head;
        printf ("The node delleted is Y.d" cur-reifo);
       free node (cur);
       return head;
 3
NODE dde orlete_rear (NODE head)
   NODE cus, prev;
    if (head + rlink == head)
      psint[" The DLL is empty \n");
      return need;
     cur= head > llink;
  prev = cur -> Wink;
    psev -> rlink = head)
     nead > llink = prw;
     freunode (cur); plints ("The nock deleted is xd" wrige);
     return head;
NODE injert left per (int item, NODE head)
   NODE temp, ur, prev;
   jut jour 2;
   if ( head A -> r Link == head)
     printf (" List empty \n");
      Netrum head;
```

```
us=head-rlink;
       while (ur!= head)
          if (idem == cur-) (Bringo) breeks
          aur= ur - rlink;
      if (ur== head)
          print ("Key not found \n");
        return head;
      prev = cur -> Llink;
      temp = getnode();
      printf ("Enter towards left of 66 /d!", item);
      scanf (" 1. d" & itema);
      temp - info = item2;
      psev -> rlink = temp;
      Jamp - Mink = prev;
      wr -> Llink = Jemp;
      demp + i link = wi
     tetum head;
NODE insert right pos (int item, NODE head).
  NODE temp, cur, next;
  itat item 2:
  if ( head -> r link == head)
  printf ("List Enopy In");
    return head;
  Cur = head -) Ilink;
CLASSMAte
```

```
while (wr!= head)
     16 (item==cur-info) break)
     cur=cur + rlink;
if (wr == head)
printf (" key not found \n");
return heed;
   next = cur > rlink;
temp = get node ();
 printf ("Enter towards the top Right of Y.d: ", item);
  scanf ("Xd V ditema);
 temp-) into = itema;
  next -> Wink = temp;
  temp → hlink = next;
  top ur -> & link = temp;
   temp + llink = aus
  return head;
NODE delete all Key (int item, NODE head)
 NODE prev, cur, nest;
   int count;
  if (head-rlink== head)
      Print ("The Del is empty \n");
      return headi
   count = 0;
   cur = head - rlink;
      CIRSSMATE
```

```
while (cur! = head)
          if ( idem! = cur > &info)
              cur=cur =rlink;
          else
             count ++;
             prev = cur - blinks
             next= cur->rlink;
             pler-> flink = next;
             next -> llink = previ
            free mode (wr);
             cur = next;
     if (count == 0):
      & printly (" Keynot found \n");
       print ("key found at 1-d positions !! and are deleted", count);
     return head;
void fearthly (int key, NODE head)
NODE temp, cur;
if (head-) rlink == head)
  printf (" list empty \n");
     retur >
  cur= head-rlink;
  ustile ( fur != Kear)
```

```
while (us! = head)
if (us → info==Key)
 printf (" Farch Succonfulta la");
  returni
     Curz Cur -> rlink;
  printf (" Lorch is not successfull ha");
   return;
 NODE ddelote duplicates (ind item, NODE hood)
  NODE prev, cur, next;
  int went = 0;
  if (head-rlink==head)
printf ("List i empty \n");
return headi
cur=head → hlink;
  while ( cur! = head)
 if (us - info!= item)
       cur = cur -) rlink;
  else
  (++tuna)
       if ( count == 1)
         cur= cur- rlink;
          Celes continue;
```

```
cloc
         prev = cur-llink;
         next = cur > rlink;
         prev-> rlink = next;
         next -> Llink = prev;
        flee (ur);
        cus = next;
if (count == 0)
    printf ("No such file item found in the list lo");
printf (" Removed all the duplicate elements of the given item
        Successfully m");
 neturn head;
void display (NODE head)
   NODE temp:
  if ( head - slink == head)
     print ("The DLL is cupty \n");
     return;
 printf ("The contents of the DIL & are: \n");
  temp = head=rlink;
  whale ( temp! = head)
     print(" "d \n" jump > info);
      temp= temp > rlink;
  printf (" \n")
CLASSMALE
```

```
void main()
   NODE head, lost;
   ind item, dioice;
   head = get node();
   head -> Islink = head;
   head - Hink = head;
   Pal (33)
     printf ("In I: Typert front In a: Injert Rear In 3: Fillele front In
              4: Delete pries In 5: Tresert key 6ft In 6: Insert k
              Right In7: Delete all Key In8: Searchitem In
               9: Delete Deplicates In 10: Display In 11: Fait);
     Scanf ("Y.d", Ochoice);
     switch ( woice)
       ose 1: Print ("Enter The item at front end: ");
               sauff ("/.d", liten);
               lost = dingert_floot(itum, head);
               break;
      case a: printf ("Fritisthe item at read end: ");
              sconf ("x d" ditem);
               last = dinsert-frees (itum, head);
               bleak;
     cox 3: last = ddelete room (head);
              bleak;
     cope 4: last = ddelete head (nead);
               breaks
     Case 5: printf (" Enter the Key item: ");
             Scant (" Y.d , & item);
             head = insert_left pool item, head);
             break;
    (age 6: printf (" Enter one key item.");
             scarf ("/d", ditar).
             head = injut - right pos ( item, head);
             break;
classmate
```

```
cose 7: printf ("Enter the Key iden: ");
         Scanf ("y.d" ditan);
         head= delete_all_key(item, head);
       bleck;
 case 8: printf-(" Enter the Key item: ");
       scaref ("/d", ditem);
       searching (item, head);
 cose 9: printf- (" Enter The Key item: ");
     scanf (" > d", ditem);
       heed = ddelete deplicates (item, heed);
       breek;
 case 10: display (head);
    breek
defautt: return;
```