```
#include estdio-hs
 # include < stalib.h>
  Struct node &
     int data;
    Struct node * next;
    struct nock + preu;
 type def struct mode + Node;
Node get rede () 2
  Node 20;
  2 = (Noele) malloc (stract size of (struct mode));
 if ( a== NULL) {
       Pf (" Nem full in");
     exit (0);
return a;
```

```
insert Front (int item, Node head) &
   Node temp, cur ;
   temp = get Nocle();
  temp as dala = item;
  cur > head > next
   head -> next = temp;
   temp -> prev - kend;
   temp-) next = cur;
   cur > prev = temp;
   return head;
Wode (rusert Rear (intitem, Node hand) ?
    Node temp, cur.
    4emp = get Node ();
   temp -> data = item;
   cur = head > prou;
  temp -> next = head.
  temp -> Prer = cer;
   cur -> ne nd = ten;
  head -> preu = +emp;
  rectum head;
```

```
Node delette Front (Node head) {
         Mode ceur, nest;
       it ( head -> next == head) {
           Pf ("Emply 'n" Fan);
           return head
     Cur = had > next.
    next = cuer = ment;
      heed > ne set = noget;
      neset -> prev = heed;
    Pf (" deleld node unith data "d", cur > data)
    free (cur);
    return had;
Woole delete Reas (Node head) 5
       Node cur, prou;
      if ( head-> moset == head ) &
            Pf ("List empty In");
           return head;
    cur = heed -> pron;
    preu = cur-> preu;
    heal spren = pren;
were sne at = head;
```

```
pf (" deleted made in the dated : " d", an sais);
 free aur);
gaturn head;
made insert Before Roy (int item, int Key, Needo
                        head ) 5
     Made tempi cun;
    temp = get Node ();
     temp - date = item;
    temp > nest = Well;
    temp -> prev = NULL;
    cur = heade - next;
   while ( cur != head) {
       if (aur -> data = = key)
          lireak;
      cur = cur - no at;
 if (cur = = head) 5
      Pf ( " item not found in) g
      return head
eur > preu -> nent = temp;
 temp > prev = cur > prev;
 cur -> prev= temp;
 temp > next = cur; return head; ?
```

```
Made insert afterkey ( int item, int key, Nade kand)
         Mode temp, cur;
         tem p= get Nede ();
        temp - data = item;
        temp > neset = NULL;
       temp -> prov = NULL;
      cur = had > next;
     cuhile (cur!= heed) q
          if (cur) data = = key) 9
       Cur = aer -> next;
  if ( cur == head ) {
             Pt ("key not found (n");
           return heed;
cur > next) prou = temp;
temp -> nent= coa-> nent;
cur = next = temp;
temp > pred = cur;
  refun head;
```

```
Node Search (ind item, Nade head) ?
     Mode cur
     int count;
      If ( head -> nond = shored) &
          Pf ("cmpfy (m");
         refurn head;
    cur = hed > next;
   while (cuer! = head) &
         if ( item ! = wer ->data) g
               cur = cure -> next;
        3 else s
             Pf ("search successful (n");
              vetern head;
    Pf (" Search un successful 'In");
    meturn head;
           my = mile many
```

```
Made delete Duplicates (intidom, Node heard) 3
    Made prev 1 cur, next;
     int count =0;
    if ( heed -> next = = heard) 5
        Pf(" list is empty (n");
       return head;
 cur = head - ment;
while ( ar! = heed ) 5
      if ( cor -) data ! = itam) 5
          cur = con -> nent;
  else g
     count ++;
         if (count :==1) 5
             Cur = cur -> ment;
            continue;
      else Epron = cor > pron;
            next = cox ment;
            iprev - neat = nent;
           Prest -> preu;
          free (cur);
         ccer = next;
```

```
id (coent ==0) 5
    Pf (" No Hem found m");
 3 else {
pf (" pernoue all dupes \m'');
 3 plur or head;
noid display ( wode head ). 5
    Wode temp;
    if (head -> neset == head) &
         pt ("emply (n");
       return;
  print f (" Contents of DLL: \n");
  tomp = head -> next;
  cubile (temp! = hend) {
            Pf (" Y.d", temp > data);
           temp = temp > neset;
    pf("\n");
```

haid main () & ind chaice , item , flag=1 , key; Made head ; head - get Node(); head > next = head; head - Tpreu - head; while (Hog == 1) 5 Pf ("Vm1. Insert Front \n2. Insert Pearla 3. Delete Front In 4. Delete Rear ms. Insert Bløre key in 6. Insert Afterkey in 7. Search in 8. Delete Duplicales in 9. Display In 10. Exit (n"); iff (" Enter chaice: "); Scorf (" r.d", & chaice); Switch (chaice) & case 1: Pf (" Enter item : m"); scorf ("'d"; & item); head = insert Front (item, head ); lireak;

```
case 2: printf (" Enter idem: \m");
         searf ("Y.d", Eitem);
        heed = insert Pear (Hem, head);
        break;
     3: had = delete Front (head);
        De lereak;
ase A: head = delete Rear (head);
        wreak;
       Pt ("Enter item: In");
case 5:
          8conf ("Y.d", Sitem);
          Printf ("Enter key: \n");
         8 conf (" /.d", & key);
        break
         heid = insert Before key (item, key, head);
 case 6: printf ("Enteritem:\m");
       & conf (" 1.d", & item);
       print f ("Enter key: In);
      s conf (" y.d", & key);
      head = insert After Kuy (it em, Kuy, head);
      loreak;
case ?: Printy ("Enter key to searchin);
      & canf ("1.d", & Key);
     head = Search (item, head).
     break;
```

```
code 8; printf ("Inter bey: (m");
       sconf ("1.d", Krey);
      had = delete Duplicates (Kay, head).
      loreak;
case 9: display (head);
       loreak;
default: exit (0);
```

Scanned with CamScanner