# include < stdio. w> # include < st dibon> node f

not info;

struct node \* rollink; \* Alink;

getnode() {

NODE \*;

X = (NODE | number(size f (8 trust node));

X = (NODE | number full (n'));

exit(o);

return x; Stanct node & janisse \* skrieter \* skou junt? typedef struct vode\* NODE; NODE getrade() & return x; NODE dinsent front (int item , MODE MODE temp, con; ; () about = getwodi (); temp > info = itum; temp - 1 Dink = NULL;

temp - Drillink = NULL,

cu = head - orline;

```
head - soline = temp;
     temp + alline = Good;
     temp - relian = cm;
      con - Diux = temp;
      return head;
NODE dinsent near (int item, NODE Pread)
    MODE temp , cm;
    temp = getuode ();
     temp = info = item/
     temp - Dhink = NULL;
     temp + 21 Dluk = NULL)
     cur = head = live;
      head = Dink = temp;
      temp - silve = head;
       cm + relieux = temp;
       temp -) Diva = cur;
       return tread;
NODE ddelek-front (NODE head ) ?
    NODE cw, wext;
     if (head - rollux = = head) &
          printf (" List is empty (");
          neturn head;
```

cur = head - rline; next = cm - xline; head & rlink = next; next - ) Deline = head; prinf!" Hen deleted at the front end (s: y.d", cur infoli free (cm); return head! NODE gggeff - real NODE prend) { MODE cmibren, if ( fread -) relieur = = fread) ? printfl" List is empty \n"); retur head; cw = head Allink; prev = cm > lliuk; prev Frlink = fread; head - Iliux = prev; printfl" I tem deleted at the rem end ic 'I.d'u', ansing); free (cm); return head; 1 oid ddisplay ( NODE head) &

```
if ( head - relieux = = head { { printf(" List is surphy [u");
 prinf ( The contents of the list are: \u");
  temp = head -> xlink;
  while ( temp! = head) &
     print[" v.d v."; temp -zinto);
      temp = temp -) silling;
desearch (int Key, NODE head) {
MODE cm;
 if (Read -) relink == head) &
 int count;
     printfl" List is supply lu"!
   cur = head - solive;
   while ( cmi! = head It cm-sinfo! = key
   count = 1;
      cm = cm - ) 2 Dink;
     count ++;
    if ( cur = = head) &
```

```
print (" Search Unsuccentull 14" ) 8;
     printfl" key element found at the parish'on I.d \" count!
   dinsat-leftpos (int item, NODE Read) $2
    MODE cm, pren, temp;
    if ( head -) reline = head) {

print[" ist is empty m"];
         retinu head!
cm = head - solink)
orlyiel cm ! = head) &
    if (cm > info = = item) {
         break)
      cw = cw-)zlink;
 if ( cm == head) &
      print{(" No Such item found
                                       in the
                                       list In");
       retur bread;
prev = cm-, llive;
: 1 shouts & = quet
```

```
temp - Mink = NULL.
printfl" Eula In item to be insated at the prints"!
temp - reliar = much
 Scarf (" V.d", & temp sinfoli
  bren - rejux = temb!
  temp + I link = cur;
   cur - Illink = tempi
    neturn head;
 NODE dinsent-rightpos (int item, NODE head)
     HOOF temp , cm, next;
      if ( head -) alink = = tread) {
printf(" List is empty \u");
             return head;
         cur = head - solive;
while ( cur soints!= head soun) &
             if( cm -) info = = item) {
break!
             cm = cm -> 8 Diak)
       if (cm = = head) {

printf (" No Such
                                   item found in the list (");
```

```
return head;
next = cm -1 soliun;
temp = gotnode ();
temp = Dink = NULL;
                        to be installed
temp - Mlink = NULLY
printf ("Enter ten ;tom
                        sight of the given
        inserted at the
       itan: (");
 scouf (" Yd", & temp + infoli
 cm - solve = temp;
 temp - Dink = cur;
  next - Ilink = temp!
   tent = wills + finst
    return head;
 NODE ddelek-duplicates l'int item,
                            MODE fread). 5
   MODE bren, cm, next;
    (int count = 0)
    if (head - soline = = tread) f
         printf(" List is empty lu");
          setur head;
```

```
Cur = head - rellink;
while ( cur! = Read) &
     if ( cm > info ! = i tem) {
          con = con-) solives!
   else f
         count H;
         if (count = = 1) f
               cm = cm -12 link!
                confinu
           pres = cm -) Dink;
            next = cm - slink;
            prev = rlive = vext;
            next of Illink = previ
            free (con);
            cu = vert;
    printfl" no such item found in the Dist \n")
if(count ==0) &
 else f
```

printfla Removed all the duplicates elements of the given item successfully hill; neturn head; ful main(15 } NODE head; int item, choice, vey; tread = get vode (); head = llink = head; fread - relieve = head; printf (" \n 1: dinsent front \n for list to 2: Linsent rear In 8. ddelpt front Inu. ddelek-rem Ins. ddisplay w6. dseuch w7. dinsut left postu 8. dinent right postu 9. ddelet duplicated in 10 exters 8 couf("1-d", & choice); switch (choice) { Care!: pringl" Enter the item at s couf (" Y.d" & item); head = divent - front (it on a head);

Cose 2: prints [" Enler the item at from end: 8 conf (" r.d", & item); Read = disat- sear (item , Read); (ork 3: Read = oldelek-front (head); Case 4: head = ddelek - near (head); break! Com 5: ddisplay (head): Cose C: printfl" Enter the Key Element to be ser ched : \"'); s conf(" ! d ", & vey); I south ( key, head); printfl" Enter the Key element n' Carl 7: scouf(".1.2", 2 key); head = divsat-left pas (key, head) break;

Case &: print (" Enter the key element: "");

s conf (".d", & key);

head = direct - right pos ( key, head;

head = direct - right pos ( key, head; preaki Core 9: printflu Ender the Key clement whose duplicates should be removed: \n'); Scanf(" -1,7", & Key); Scouf(" Nd", & key);

head = ddelek\_duplicates (key, head)

break;

exit(0); défault: exit(0); return o;