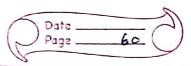
Shrishtha Aggarwal 1-BM19CSISS

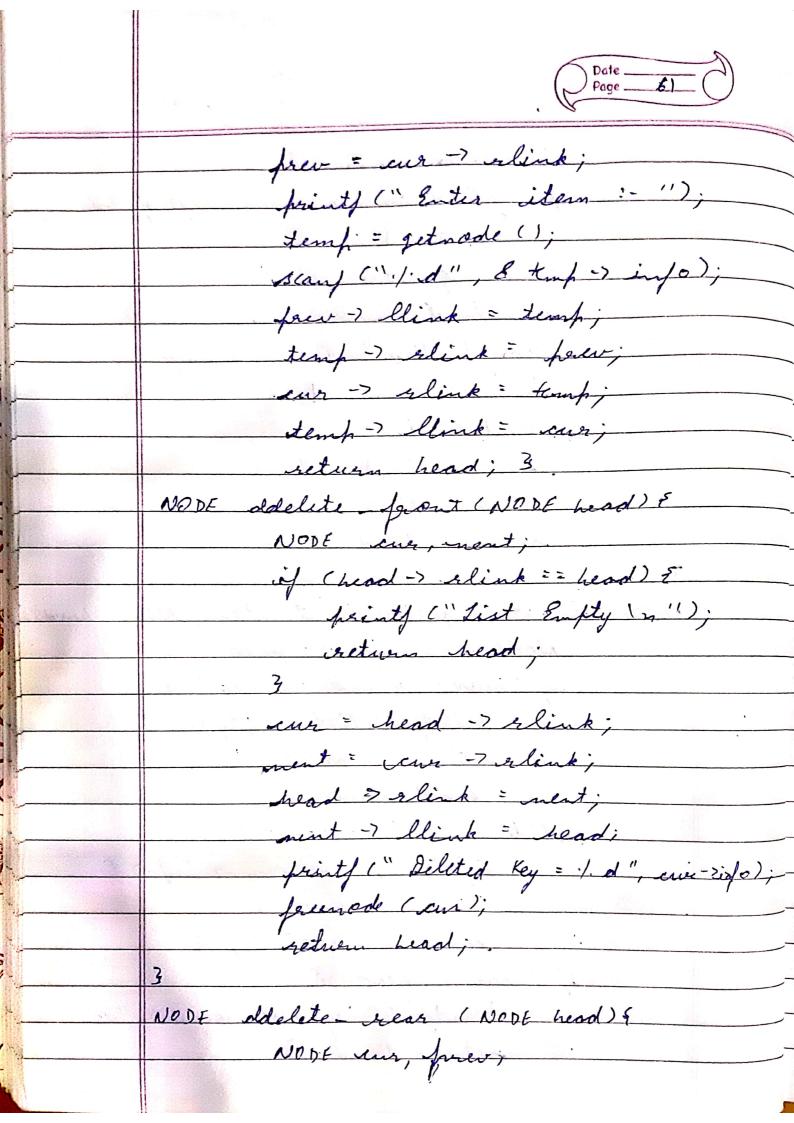
Date
Page 58 Lab - 9 (Week 10) 14/12/20 do with of sloubly linked bet : g creation s) insertion (a) delete front d) delete evar e) Search 1) insurtion before & after key of Delete refeating occurences # include = stdio. 4> # include < stdlib. h? staret nade & int info; struct node * slink; struct node " Clink; 3; typedel struct node + Node; Noole getnode () & Node x; n = (Node) malloc (sysof (struct mode)); if (n = = NUIL) & frients ("Tremory full 12"); drit (0); 3 return x; 3 void freenode (Node u) & free (u); 3 Node dinsert rear (Node head, int item) { Node they sur; Hemp = getnesle (); temp - 2 wlink = NULL; temp -> llink = NULL; temp -> info = item; eur -? rlink = temp; head -? llink = tent; temp -> rlink = head;

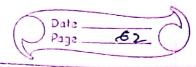


head - info = heard or rinfo 11; return head; 3 dinsert - from (int item, Abde head)? Node timp, eur; demp = getnode (); Icuh - 7 info - item, cur's head -7 Ilink; head - ? Ilink = temp; temp ? llink = head; temp -> rlink = euri; cur -> clink = temp; seturn head; Node insert lefthos (int item, Node head) & Node temp, our, freez; if (head -7 rlink = = head) & frint (" List empty in"); return head; cur = head => rlink; while (ur! = head) & if (item == ur-rufsur = en -7 rlink; 3 if (con == head) & fait (" Not found x") return head; 3

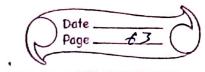


free = cur -> clink; prints (" Enter towards left of of.d = " item); temp = getnode (); scanf - to getonate scarf ("i/id", & temp > info); free -? Elink = temp; temp > llink = freer; eur -> llink = temp; Lump -7 Rlink = ens; return head; NODE insert rightfor (int item, NODE head) { NodE temp, cur, prev; if (head -> rlink == head) { prints ("List empty (n"); return head; } sur = head -> rlink; while (cur! = head) { if Citem = = «un - > info) & break; 3 such = sur -> rlink; } if (cur = = head) & prints (" Key Not found \"); return head; 3



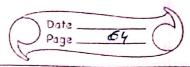


if (head > wlink == head) { prints (" List empty (""). seture head; 3 eur = head > llink; frev = cur -> llink; head -> llink = prev; frev -> alink = head; freemade (com); void display (NODE head) &. NODE temp; if (head > wlink = = head) { frients (" List empty .. ("). Ilturn; 3 for (temp = head > rlink; temp! = head; temp = temp -) slink) & frints ("/d /m", temp -> info); 3, word search (int item, NODE head) & NODE temp; int j = 0; temp = head -> rlink; while (temp! = head) & j++;



if (temp > info = item) & frints " I tem at 1.d" , j); return; temp = temp > slink; 3 frients (" Search Unsuccessful "); 3 NODE delete all (intitem, NORE head) & NODE puer, lur, ment; int count = 0; j=0; if (head > elink == head) { frients (" to Not found"); return head; 3 eur = head - irlink; while (seus ! = head) } if (item : cur > info) { en = cur -> rlink; } else & j++; if (j == 1) { cus = cus -> chis} if (item= = cur) info } ? count ++: frer - cur > Slink; ment = en - rlink; free -> elink 2 ment; ment I lik & prev; freemode (com);

cur = mend; 3 3 3



frit ("No duplicates \4"); 3
else frints (" Ref. Key at 1. d & del," (out); int main () { int item, chaice; NODE head; head = getnodo (); head - rlink = head; head - llink = head; for (;;) & frints ("In 1 Insert reas In 2 Ensert front-1 3. Insert left in 4. Insert right in 5. Delete front 1 5 6. Delete rear 1 7. Delete dup \n 8. Search \n9 Display "); frients (" Enter choice: "); scanf (" . / . d", I showe); switch (choice) { case 1: freint ("Enter item: "); scanf (" / d", & item); head = dinsert rear (head, item); break; case? : frints ("Enter iden: "); scanf (", d", & item);

