22/1/24) Wap to implement Singly linked list with following operation: a) Create a linked list b) Invertion of node at first position, at any position and at end of list c) Display the contents of the linkel > #include (stelio h) # include (math.h) # include < string. h> struct node { int data; struct node * neut; Street node * counte Node (Int data) { Struct node * new Node; newMode = (Stand node") mallac (Size of (Stand node)) of (new Mode = = NULL) { print ("Memory allocation failed"); newMode > data = datas newplock > nent = NULL; return new Norte

struct node * cr-ll (int value [], int-size) { Struct node * head = NULL; Struct made * tail = NULL; for (int i=0; i < size; ++i) { Struct node new Hode = exceptifiede (Values [i]); if (head == NULL) & head = new Mode; tail - pershlode; else E tail > next = new Node; tail = new Mode; return head; 1) void insert First (struct node ** head , jut data) { struct node * new Mode; new Mode = create Mode (data); newNode -> nent = * head; * head = new Mode;

```
void insert At hos ( struct node * * head, but date
            int hosition) {
     if ( hosition == 0) {
        insertfirst (head, data);
  struct node * new-node = create Node (data);
struct node * current = * head;
    for ( int i= 0; i < position -1; ++i) {
      frint ("Tovolid hosistion In");

potum ;
   Current = current -> nend;
     new-node -> nout = arrent -> nent;
  Current - new node;
 void injusted ( stant mode * * head, but datal
    struct mode * new node = create Node (data)
         if ( * head = = NULL) {

* head = new-node;
```

struct node. * current = * head; data while (current -> nent != NULL) { Current = current > neut: current -> nent = new-node; void display (struct & node thead) { while (head != NULL) {

printy (" 1.d, ", head > data);

head = head > next;
} Jeint (" (n"); 3 () main () { rut values = {1,2,3,4}; struct mode * linked list = metalistellist(); just data; fright (Enter date to just at the beginning); ingest First (& linked List, data); data) { frints (" Enter data to insert at specific thesition: ") scenf("1.d", belata);

frinty ("Enter the position: "); scang 1" 1.d", & position); insurt At pos (bolinked list, data, position); fring ("Enter data to insut at end: "); scory (" ".d", & data); insert End (A linked list, data); display (linked list); oreturn 0; Output: Enter the number of do clements: 6 Enter data to insert at the beginning: 58 Enter chata to insert at specific presition: 3 69 Enter data to insert at the End: 100 50, 1, 2, 3, 4, 69, 5, 6, 100 NULL 2) WAP to implement singly Linked hist with following operation: a) Create a linked list b) Deletier of first element, specific element and last element in the list. e) Display the contents of the linked list. # include (stdio.h) # include (Stalib.h) Struct node { int data; struct mode *next; stant node * condtentale (int data) { Struct node * new Node = (struct node *) malloc (size of (struct node)); if (newflode - = MULL) {

printy ("Memory allocation failed in"); exit(1); men > made new Mode - data = data; new Node > nent = NULL; returno new Node;

struct node * smattle or-UL) { Stewd mode * head = NULL; struct mode * tail = NULL; int size, clata; . points (" Enter the no of elements "); scanj ("1.d", & size); fronty (" Enter the elements In");

for (int i=0; i < size; +ti) {

scary (" '/.d", & data); struct node * new Mode = create Mode (data); if (head = = NULL) { head = new Node; tail = new Moch; tail-> nent = new Node; tail = new Mode 3 return pead:

```
void date deletetierst (struct rode * head ) {
       H ( * head = = NULL) {
        fring (" List is empty. Nothing to delete ");
       return;
   struct node * toup = * head;
   *head = (*head) > next;
  free (temp)
 void detetef lement ( struct Mode * * head, jut key) {
      if ( * head = = NULL) {
     points ("List is Empty. Nothing to delete.");
     struct node * current = * head;
     struct node * prece = now NULL;
     while (current != NULL L & current > data!=
          key) {
        free = current;
      current = current -> next;
       if ( current == NIULL) {

frints ( " Slement not found in the list ");
```

```
of (free == NULL) {
     * head = current > next;
   the {

free > nent = current > nend;
}
  free ( current);
void deletelast ( struct node ** head) {
private ( "List is empty. Nothing to delete.");
 if ((*head) -> nent == NULL) {
     fre (* head);

* head = NULL;
 struct node * current = * head;
 struct node * how = MULL;
    while (current + nent ) = NUH) &
         first = current;
          current - current - nent;
    prece -> new = NULL:
    Jue (current):
```

```
void display ( struct node * head) {
     while ( head != NULL) {

prints ("1.d,", head > data);
   head = head -> neut;
 Specinty (" NULL IN");
 jut main () {
      struct mode * ll = cr-ll();
     frint (" Linked list before deletion: "):
     display (ll);
      delete First (&ll);
      deletetist (till),
frivil ("Linked list after deleting first element ");
     display (IR);
      int key;
  perint (" Enter the climent to delete ");
  scump ( "I.d ", b key);
   frints 1" Linked list often deleting specific element In");
   display (ll);
    delite ast ( & ll);
printy ( "Linked list after deliting last element In");
    display (ll);
   seturn 0;
```

Output Enter the number of clements: 6; Enter the elements: Linked dist before deletion: 1, 3, 4, 5, 6, 2, NOLL Linked List after deleting first element: 3, 4, 5, 6, 2, NULL Enter the climent to delete: 4. Linked dist after deleting specific element: 3, 5, 6, 2, NULL Linked dist often deleting last element: 3,5,6, NULL.