1) Implement Cherogram to perform insertion operation (on all the three position on singly linked list, WAP to Implement Singly Linked List with following operations a) Create a linked list b) Insertion of a node at first position, at any position and at end of list. Display the contents of the linkedlist # include <stdio. h> #include <stdlib.h> typedef struct Node {

int data;

struct Node \*mext;
} Node; void Insert At Beginning (Node \*\* head ref. uoid Insert At End (Node \*\* head ref, int newdata);
void Insert At Position (Node \*\* head ref) int newdata, int pos); void Printhist (Node \* node) void InsertAtBeginning (Node \*\* headeret Node \* neurnode = (Node \*) malloc (sizeof (Node);

neuenode > data = neuedata; new node > next = \* head ref; \*headeref = new node; void Insert At End (Node \*\* head ref. Noole \* neveroole = (Noole \*) malloc (size of (Node)). Node \* last = \* head ref; newprode > data = newdata; if (\*headeref == NULL) \*headref = newprode;

oreturn;
} nohile (last -> next) = NULL)

last = last -> next;

last -> next = new node;

3 void Insert At Position (Node \*\* head refs int new data, int pos) if (\*headeref == NULL && pas >1) printf ("Cannot insert at the specified position in");

return;

Node \* temp = \* head ref; Node \* neudlode = (Node \*) malloc (size of (Node) new Node -s data = new data; newNode > next = NVII; noticle (-- pos >0) & if (temp == NULL) ? froints ("Cannot insert at the specified position in"); temp = temp > next; if (temp == NVIL) {

forintf("Cannot insert at the

specified position \n");

oreturn; newNode -> next = temp -> next; temp -> next = newNode; Perinthist (Node \* node) { hehile (nodel = NULL) {

frintf ("%d \n's node > data). node = node -> rext; int main () { int cho news pos; Node \* head = NULL'

while (ch]=5){ pounts ("Menula"); frintf ("1. Insert at beginning \n"); prints ("2. Insent at a specific paintion in").

prints ("3. Insent at end in"). pointf ("4. Display linked listin"); points ("Enter your choice)"). scaref ("% &ch);
switch (ch) { prints ("Enter the data you want to insent at the beginning in") scanf ("% od" & new); Insert At Beginning (Shead, new); points ("Enter the data and position scanf (" % of % od" & news, & pas); Insert At Position Shead, new pos); freints ("Enter the data you want to insert at the end in") scanf ("/od", gnew); InsertAtEnd( & head, new); freintf (" Created linked list is: 'n"); Perintdist (head); break;

case 5: return 0;

default:

fruitt ("Invalid choice! 'n"). eretuan 0; Output: 1. Insert at beginning 2. Insert at a specific position 3. Insert at end 4. Display linked list 5. Exit Enter your choice Enter the data you want to insert at the beginning 1 Insert at leginning 2 Insert at a specific position 3 Insert at end 4. Display linked list 5. Exit Enter your choice Enter the data you want to insert at the end 2. Insert at a specific position 3. Insert at end 4. Display linked list 5. Exit 1. Insent at beginning Enter your choice Enter the data and position at which you want to insert 1 Insert at beginning 2 Insert at a specific position 3 Insert at end 4 Display linked list Enter your choice Created linked list is: 1. Insert at beginning
2. Insert at a specific position 3. Insert at end 4. Display linked list 5. Exit Enter your choice 5

WAP to Implement Singly Linked dist Dieth following operations

Directe a linked list, 6) Deletion of first elements specified element and last element in the list Display the contents of the linked #include <stdib.h> typedet struct Nodes

int data;

struct Node \* next;

} Node; void Insert At Beginning (Node \*\* head nep int newdata); void Delete At Beginning (Node \*\* headeref); void Delete At End (Node \*\* head ref); void Delete At Position (Node \*\* head ref, int pos) void Perintdist (Node \* next); void Insert At Beginning (Node \*\* head ref. Node \*neverade = (Node \*) malloc(size of (Node) neronade -> data = nevedata; newnode > next = \*headref \* head ref = neverode;

void DeleteAtBeginning (Node \*\* headreps Node \*ptr; Node \*ptr; if (\*head nef == NULL) frintf("\ndist is empty"); else É ptr = \*head nef; \*head nef = ptr > next; free (btr);
freintf("In Node deleted from the
beginning..."). void Delete At End (Node \*\* head ref) ? Node \*ptr, \*ptri;
if (\* head ref == NULL) forintf("\nlist is empty")
else if((\*head ref) -> next == NULL) { free (\*head ref); \* head ref = NULL; printf (" in Only node of the list deleted..."); ptr = \*head nef; while (ptr -> next! = NULL) { ptr = ptr;

ptr = ptr > next; ptrl -> next = NULL; free (ptr); frints ("In Deleted Node from the last.");

void Deletest Position (Node \*\* head refint Node \*temp= \*headref , \*prev; if (temp == NUI)?

printf ("Indist is empty"); return; of (pos = = 1) { \* head ref = temp > next; free(temp);

freintf ("In Deleted mode north

position %d"; pros);

return; }

for (int i=0; temp! = NULL & i < pas-1; itt)

forew = temp;

temp=temp;

rent; if (terrif==NULL){

printf("nPosition out of range"); return; } prev -> next = temp -> next; foree (temps); forint ("In Deleted node with position % d", pros); void Perintdist (Node \* node) { while ( mode /= NULD ?

forint f ("co/od In", node > data); node = node - next;

int main () { int ch, new, hos: Node \* head = NULL: while (ch != 6) f printf (" InMenu In"). pointf("1. Create a linked list In"); printf ("2. Delete at beginning In")? printf ("3. Delete at a specific positioning).

printf ("4. Delete at end in").

printf ("5. Display linked list in").

printf ("6. Exit in"). printf ("Enter your choice"); Scanf ("hd "seh); switch (ch) { case 1: prints ("Enter the data you want to insert at the beginning "); Scant ("Ld", &new); Insert At Beginning (Shead, news); case 2: DeleteAtBeginning (&head); scanf ("Fod"; & pos);
Deletest Position (& head, pos); case 4: DeletestEnd(4 head); break; Case 5: frintf ("Created linked list is: 15).
Pointhist (head); break; sase 6: return 0;

3 default: printf ("Invalid choice! 12) I oretworn O's Output: Menu 1. Create a linked list 2. Delete at beginning 3. Delete at a specific position 4. Delete at end 5. Display linked list Enter your choice Enter the data you want to insert at the beginning 1. breate a linked list 2. Delete at beginning 3. Delete at a specific position 4. Delete at end 5. Display linked list Enter your choice Node deleted from the beginning