```
Lest Counting LL,
                     Linked
                                              element
                              Trenting
  # Societa (stolo.h)
  # Soclude (malloc.h)
   stauct node
      int data;
      stauct node + ment;
 stauct made + stoot = NULL;
         node + concate_ll (stouct node+);
         node + display (stout node *);
 Howet
         node + Insent beg (stouct node )
       node * Intent_end (stouct node *);
 stouct
stauct
       node + Insert - before (struct node +)
         node + Insert-after (stowet node );
stouct node * snort_list (stouct mode *);
stauct node + sont list (stauct node +).
int main () {
 Int choice;
 point ( In ). Oceate a list. In 2. Sisplay in 3. Ensent beg
        in 4. Insert_end in 5. Insert_behave in 6. Insert_ofter
  2005
  pf("In Enter the choice:");
  of ("1.d" & chosce);
  switch (choice) {
     case 1: start = coleate_ll(start);
             paints (" Ht coeasted");
              beleats;
    Case 2: start = desplay (start);
           bounds;
```

```
case 3: stoot = insort beg (start);
         bereaks;
case 4: start = Ensent_and (start);
         break;
case 5: start = sout list (start);
         becah;
Zukle (chosce !=6);
 section o.
stouct node + counter-U Cotouct node + start) &
   stauct node *new_node, *ptai;
  int num;
  possitife" Enter -1 to endit;
  pf (" In Enter the date");
  SHC"+d", (y Jata);
  while (num ! = 4) }
      new-rode = (stauct node *) malloc (stree of (stauct node))
      new-node >> data = num;
    of (Start = = NULL) &
      new-node -> next = NULL;
        start = new_node,
    relses
         pta=stant;
          while (pton-> next != NULL)
            pton = pton-next;
        pton -> next = new node;
```

new_node -> Next = NULL;

```
poshtf ("In Enter the data");
    stant ("+d", gnum);
  section start;
 stouct node +dsplay (stouct node *start)
  stauct node apter;
   Ptge = stant;
   whole (pto = ! = NULL)
     Pt(" It 1d"; pta data);
     pta = pta - next;
   sections start;
stauct node * Insert-beg (stauct node *start)
 stauct node + new node;
 Int num
 of ( enter the deta");
 sfc"+d", g num);
 new node = (stouct node +) malloc (stree of (stouct nod))
 new anode => data = num;
 new-node -> next = start
 Stoot = new node; have the
section stort;
                 real made - made - where con
```

```
stauct node of Essent and (stouct node "start)
  stouct node * new - node, * pto;
    sot ours;
   posint (" to enter the data!");
    sconf (" 1 d", & num);
   new_node -> data = num;
   new-node -> next = NULL;
  pton = sterrt;
  of CARDA == SURVEY
       start = new-node;
   7 elsex
       whele (ptor->oest! = NULL) {
           Ptor = pton - next; }"
      pton -> next = new_node;
      Heturn start;
staut node * Insert_before Citymet node * stout &
   stamet node * new node, *ptor, * pouptor;
   306 num, val;
   Pt (" vo Roter the data");
    Sf(" 4-d.", & num);
    97 (" in enten the data before which value to be
          inserted");
   St("1.2" b.1")
   new-node = (stomet node*) mallor (street Cstonet node)
   new_node -> data= num;
   ptn = start;
```

while Coten -> datoy = nal) & pocapton = pton; ptn = ptn -> next; Sot temp; pocuptor > next = new nade; new-node -> next = pto1; setum start; Stouct Rosert-after (stouct node *start) & stouct node *now node, *pton, *peupton; fort val, num; pt(" youter the data"); SF ("Hd", & num); ptUm enter the value after which data to be my -fed"); sf0"1.d", & val); new_node = (stouct node *) mallow (size of (stouct not) new_node -> data= num; pton = start; pouptor = ptor; while (pouptor -> data !=val) perepter = pter; pton = pton - next; poseptor + next = new-rode; new_node => new = ptog setum start;

stauct node + sout_list · stouct node uptorlyptors; pto1 = start; whole Upton 1 - rout != NULL) pto 2 = pto 1 -> placext; while Cotor 21 = NULL) & if (ptn 1 -> data >pts 2 -> data) temp = ptol -> data; pto1 -> docta = pto12>dota; pto12 -> docta = temp; ptor 2 = ptor2 -> next; ptal =ptal -ment; desplay (start);

1 coucate a Linked List e) WAP to (1) Deletton of @ Planst element (Last element @ specified element # Roclude (StdRo.h) # Include (malloc. h) stauct node Int data; 3 Statuet node mout; stouct node * Ascast = NULL; Ascust node = cocatell (stouct noder) strenet node * display (struct node*) stouch node + delete beg (stouch node +); stauet node * delete-end (stauet nodest); stouct node * delete before (Street node) Stauct rode delete after (stault node"); Stouct rode & sout 18t (stouct node *); But main ()5 Port shore; lest 6. Aspley 7 sout 1836 8. exet Pfl" 1. delpheg . 2. bel and . 3. del-behove 4. del-ath Al +do, y chosce); pf (" 4 couche_LL 5. display 1. delete-beg & 2. delete_end 3. delete_rode); Sf (4. d. , & chopce);

switch (choice) s

cose 1: delete - beg (start);

abovers;

cose 2: delete - end (start);

bovers;

bovers;

cose 4: delete - after (start);

pf (" Linked List coverted");

bovers;

case 5: desplay (start);

bovers;

cose 5: desplay (start);

bovers;

cose 5: delete - list (start);

bovers;

cose 5: delete - list (start);

bovers;

cose 5: exit (o

)

while (choice != 10).

secture 0;

stormet node of coreate_U (stormet mode of stort); stand and * Laplay (stanet and + start); Stourt node +new node, +ptor; stouct hade *ptx; pton = start; fort num; while (pto) = NULL) pf (" menter - 1 to end"); pt("It ifd", plon -doda), pt (" enten the data!"); pton = pton -> next; SHU" +- d", & gum); while comm =- 1) f between start, new-node = (stouct node) malloc (size of (stouch 3; node)); stouct node * delete beg (storuct node * stort); new-node -> data = num; stanuct node * pta; it (stort == NULL)1 pto = start; new node + next = NULL; start start-ment start = new nodes else force (pton), setum start; pton = start; while (pt n -> next [ENVIL) Stouct node * delete_end (stouct node * start) & ptn = pton - next; stouct node *pton, * poceptons ptr =ment = new_node; ptn = start; while (pta -) next != NULL)+ new-node -> ment = NULL; pacepta = ptor; section start; Coton = pta -> mext; purply => next = NOLL; force (pts); Dietum Estart;

delete - node (stouct node x sto struct node * uptor, upouptor; stowet node pt (" enter the value to be deleted"); sf (" 1.d", gval); ptn = start; if (ptn + data = = 2val) } start = delete-bog (start); else f while (ptn + data ! = val) people = ptn; ptin = pton > next; (cety) and pose flor -> next = ptn -> rent; force (pton); return start;

22/01/2024 Inscriben pargonomme

O/p:-1. concate a list 2. display 3. insert-beg

4. insert-end 5. insert-bet 6. insert-after

enter choice: 1

enter -1 to end
enter the data: 1 2 3 +
enter the data: -1

enter the choice: 3
Enter the deda: a
Enter the choice: †
enter the dota: 5
enter the choice: 2
0 1 2 3 4
enter the choice 7

enter the choice 7.

3 +

-

-

Deletton perogram

op:

1. coueste a Dist 2. display 3. delete-beg 4. delete-end 5. delete - mode

ector-the children

enter the choice: 2

Enter the choice: 1

enter -1 to end

1234561 enter the shorter data:

enter the docta -1

enter the chosce &

6+ Tal set intes 1 2 3 4 5

enter the choice 3

enter the choice 4

enter the droppe 5

enter the value to be deleted. 3

enter the choice 2