```
# Include < etdio. h
 # ineludec etalib. h>
  Struct node
      int data?
     struct node knext; etruet næde prev
   væded etruct node * head;
   upid beginseet();
   woid bindenut ();
   word randeneut ().
   usid begdells;
usid begdenddells.
  word mais c
   int choice = 0;
while (choice 1 = 8)
  printf ("In 1. ineut at the beginning In 2. Ineut at the en
3. Inestat random \n4. delete at the beginning \n5. dele
at the end \n6. Delete at random\n7. Duplay \n8. Exit
 printf ("Enter any chiece:
 Scanf ( "% d", & choice);
witch (choice)
 case 1: beginsest();
 break;
 care 2: bondine et ();
  break?
care 3: randinect ();
```

```
break :
 care 4: begdell.
  briak;
 Call 5: enddel ();
  break.
 call 6: randall ();
break; case 7: dieplay();
brukk; Erêt (0);
 break;
word beginnent ()
 etruct node & pty?
 int item;
 ptr = letruct node *) malloc(eize of (etruct node);
  y(ptx == NULL)
   puntf l" In ourglow");
     purte (In Enter item value");
     scanf ("/od", &item);
    y (head == NULL)
      pty-> next = NULL;
      ptr -> paer = NULL;
      str -> data = item;
      Read = ptr;
```

Scanned by TapScanner

```
ptr -> data = tem;
  ptr -> prev = NULL;
  ptr -> next = head.
  head -> prev = ptr;
3 painté (n'node merte d').
word care end unet ()
 struct node * ptr + temp,
 int item;
 ptr = (utruct node *) malloc (lize of (struct node)
  (ptr == NULL)
 printf ("In aurflow);
 Brintf ("In enter value");
   'scanf ( " % od 1, 4 tem);
   ptr -> data = itum;
  if ( head = = NULL)
     ptr -> next = NULL',
     ptr -> prev = NULL,
     head = ptro
   elee ?
  temp = head;
   while Hemp -> next | = NULL)
   femp = temp -> next;
 temp -> next = ptar;
  seint ("In node creerted").
```

```
. store
void randineett
 int item, loc, i.
 struct node & ptr, & timp.
 ptr = letruct woder) maleoc (eize of (etruct node).
   puraf (" auflour").
    temp=head,
    punt (tinter location: ");
     scant ("bd", 4600).
    for (i=0; i< loc, i++
      temp = temp -> next o
      if (temp == 'NULL).
        peintf (" there are lives than "/- deliments"
                                          ,loc)
        etuno, mono
      print+(Frite value");
      'Scanf ("/od", (ten);
       otr-) data = jtim;
       ptr->next=temp->nexto
      pto -> prev= tempo,
temp -> new = ptoo,
      temp -> next => proex = ptro,
       printf ["In node neutid It);
   youd begdel ()
    stored node pto;
    U/ hand == N'ULL
             " underflow")
```

Scanned by TapScanner

```
Mark Command C
                           elling (head - next == NUL)
                               3 head = NULL"
                                  free (head);
                                peintf ("In mode deleted in").
                              ptor = head ;
                                   head = head-s next,
                                   head - prey = NULL;
                                   forespotos;
                                  printe ("In node deleted")"
                        void delend ()
                          struct node & ptro
                                  glhead == NUIL)

printf ("In indufeoro").

else if Chead -> next == NUIL)

head = NUIL.
                                          frei(head):
                                       printf ("In node delited In").
                                      ptor = hel ?
                                     if (ptor -> next = NULL)
                                    3 ptor = ptor = neut.
3 ptor -> prev -> neut = NULL;
                               free (ptg) -,
                          sprintt ("In node deleted In").
              void randdel ()
            Estruct node * ptr, * temp;
                     ent vale,
                printf ("Entre the location: ")
             Scant ("/·d", gral);
```

Scanned by TapScanner

```
vehile (pto -> data 1= val)
 ptr = ptr -> next "
  lotor nent == 2 NULL)
  printf (" Can't delete")?
eluig(ptx-) next-) next == null)
 temp= pto -> next;
ptor-next = proon temp - peat;
 free (temp):
 printt!" In node delited in").
Estoret pode ptro
 printe ("printing valu: In").
   printe ("% dln", ptr-) data)
    ptr=ptr-nexte
```

Scanned by TapScanner