```
# include < stdion>
# include < stacib . h>
  struct node
  s int data; book & scale
   struct node * next;
  struct node * head = NULL;
   int longth = 0;
   void insertend (int ele)
    struct node * new node, * temp;
     newnode = ( struct node *) malloc ( size of (struct node))
    nowhode > data = ele; = < 200 -
    nownode -> next = NULL; bastosia
    if ( head == NULL)
      head = newhoode / + 200 = 2000 =0000
   1 temp = temp = next;
      temp -> next = newnode;
     longth + +1
```

```
void insert front ( int ele)
  struct node * + emp;
  temp = (struct node *) malloc (size of (struct node));
  temp > data = ele;
  temp > next = head;
  head = temp;
  length ++;
void insert random (int ale, int pos)
 insert front (all)
 alse if (pos> = longth+1);
  insertend (cle);
 alse
    struct node + mst;
  eigt ( Struct node *) malloc ( 813e of (struct node))
   struct node + temp;
 temp = (struct node ») malloc (size of (struct node))
 temp: head;
for (int i=1; i<pos-1; i+1)
 { temp = temp > next;
  ing + > data = Ile;
  inst - next = temp + next,
  temp -> next = inst;
  length++;
```

```
void delete front ()
   if (length = = 0)
   { point { (" In List is empty. In");
{ street node + temp;
   temp = (struct node *) malloc ( size of (struct node));
    temp = head;
    head = head -> next;
   temp -> next = NULL;
    point (" In The element doleted is old", temp > date)
  void deletered ()
      if (length==0)
     3 prints ("In list is emply. In");
    & street node > temp;
      temp = (struct nocle to) malloc (Size of (struct node));
       temp = nead;
```

```
while (temp - next - next! = NULL)
  temp > temp > next ,
 Struct node & del;
 del = (Struct node +) matter designal
 malloc (sixe of (strut nocle));
 del = temp > next;
  temp - next = NULL;
 length - -;
 Point ("In The olement deleted is ' of od', del + data)
void delete sondom (int pos)
  if (length = = 0)
  paintf [" list is empty \n"];
 else it (pos == 1)
 delete front ();
elseit (pos) = longth + 1)
deleternd ()
olee
   Storut mode + del ;
  del = (struct mode *) malloc (size of (shut mode))
```

```
Strut node + temp;
temp = (struct node *) malloc (size of ( struct node)),
temp = wad;
 for (int i=1; i<pos-1; i++)
     temp = temp=> next;
    del = temp -> next;
  temp - next = del - next;
  del - next = NULL',
  Deugth --;
  print + (" In The element delited is : 1.d, del - data);
  void display ()
  E shut node " temp;
   temp: (struct noch *) malloc ( size of ( should noch));
   temp = head ,
   ( JUM = = que) Li
    pomtif (" yn list is emply In);
     Ilse
```

point (" In The contents of the list are: In"), while (temp! = NULL) printf (" y'd In", temp-> data); temp = temp - next; int main () { int droke, de, post from state char ch; 2); pointf ( "In 1. Insert at end In 2. Insert at front In 3. Insert at random position in 4. Despay In 5. Delete at front In 6. Delete at end In 7. Doleh at random in 8. Exit"); point f (" In Enter your choice ")" scomp (11 y.d", & show); suitar (choice) case 1. printf ("Enter the element to be ungerted In'l); scarf (',d", fell)',

```
inscrtand (ell)
Case 2: print f ("Enter the element to be insented his
      scanf (11/d", & ell);
      inscriptont (ell);
Case 3! printf ("Enter the clament to be inserted)
       scarf ("yd" , foll)
      print ! (" Enter the position (n');
       scout (11/211, 2 pos);
      miertrandom (ele, pos);
      break;
 Case 4; display ()
break;
 coure 5; delete front ()
     break;
 Cone 6: deleternal ();
      break!
 Care 7: printf ("InEnter flu position!");
     scart ["1.d", 1 pos);
     deleterandom (pos);
    } course ( angiel = 0)
    roturn 0!
```