```
#include < stdio.n>
# include < stdlib. h>
Stewet mode of
Ent Plem;
sterret mode * link?;
typedel stemet mode * NODE;
NODE getadde () {
 NODE Nº
 x = (NODE) malloc (size of (stauct noole));
 of (x == NULL)
  & perenty (" memory full");
 2 exit(0).
 setuen x"
word freemade (NODE x) {
z fuce (x).
 NODE facett front (NODE facent, Put & tous) &
 NODE temp;
  temp = getnode();
  temp > info = item;
 temp > link = NULL;
  of (front == NULL)
  { return temp; }
  temp - link = ficest & F
   foust = temp;
   autuum fierst;
```

```
NODE insent-relan (NODE finst, int Dem) {
  NODE cue, temp;
   temp = getnode ()"
   temp > info = item;
   temp > Link = NULL:
   of (filest == NULL)
    eletuem temp;
   cur = fierst :
   while ( cue > link 1= NULL)
      cum = cum > linko
     am + link = temp's
   aleturn finst "
 NODE delete-front (NODE flust) {
 NODE temp; if (finat = = NULL) {
  perint (" list is unpty");
  neturn fierst; &
 Comp = faust;
 temp = temp > link?
puint (" item delited at fuont end");
 full (first).
setuen temp
NODE delle-mean (NODE finst) {
 NODE au, puer;
if (first == NULL) {
  puint[ l'list is empty");
 gretnem first; &
```

```
of (fiest > link = = NULL)
 { puint (" "tem deleted is "/od", fiest > into);
   forer (first);
   altein NULL;
 POLEN = NULL ?
Cuer = fierst ?
 while ( cui > link ;= NULL)
   & plier = cuer;
    Cue = cue + link;
 Doubn't (" Per deleted is "od", cuer > info);
 force (cur);
 Prev - link = NULL:
 Metuen fierst;
void display (NODE final) of
 NODE LEMP
  if (fillst == NULL)
  Puint (" West is empty");
 for ( temp = foust , temp; = NULL; temp =
   temp > link) {
    pounts ("% d In", tempsinto);
```

```
NODE Ensect-pos( ent pos, NODE flust) }
 NODE temp, au, puer;
 Put count
temp = getinade ();
 temp > info = item;
temp > link = NULL;
if (first == NULL && pos == 1
 & return temp; &
 if (finst == NULL)
 & puint (" Invalid position");
   settem finst; }
 it (bor == 1)
 & temp > link = fiest;
   finst = temp;
   alter temp?
 00 mut = 10
 PRIEN = NULLO
  cuer = first :
 while (cuer's=NULL 22 count &= pos)
 & preev = aner.
   cuer = cur > linko
   count ++;
?+ (count == pos)
puer > link = temp;
temp + link = cur.
setuen filest;
```

```
perint (" Envalled position");
   autuem finst?
  NODE delete-pos (NODE flut $, int pos) }
     NODE aur
      NODE perev;
      Put count = 0, flgg = 0;
      If (faut == NULL 11 pos <0)
     I puint (" "invalid pos");
     if (pos = = 1)}
     cuer = firest "
     fierst = fierst > link;
     facenode (cur);
    neturn first .
   Decen = NULL
   au = finst
   count = 10
   while ( cur ! = NULL)
   & 4 (count == pos) } flag=1; break;}
   count ++:
   Perer = curi
   aur = aur > link; }
 perint (" invalled position"); oretrem firest; }
 perent (" Frem deleted is one ofod", cum sinto);
 pow > llnk = cum > llnk ?
freenode (au);
gretuem finst ;
```

```
P vold main () }
     int item, pos, ch;
     NODE flest = NULL,
    ton (;;) }
    pennt ("Enter your choice. 1. Insert-front
    In 2 Enseut mean In 3 delete pront In 4
     delete mean In 4 display In5 inscrit
     at specific position in 6 furest o
    delete at specific position");
   Scary ("%d",2 ch);
   Switch (un) of
    case 1: peulit ("Enter êtem to be inserted");
           scarf("% d", L'item);
flest = insert - fevent (flust, item);
            bueak;
  case 2: perint ("Enter êtem to be inserted");
           scamp ("% d", Litem);
finst = insent-snear (finst, Penn);
            buleak;
            first = delete - front (first);
case 3°
           bueak;
           finst = delete_near (finst);
case 4.
           display (first);
           break;
```

```
case 6: perint (" linka Etem to be Enseated");

Scanf ("% d", 2 item);

Perint ("Finter position");

Scanf ("% d", 2 pos);

first = insect - pos (first, item, pos);

break;

case 7: perint ("Finter Hem position to be deleted"),

scanf ("% d", 2 pos);

first = delete pos (first, pos);
```