```
#include < stdio. h>
Stauct node {
     int value.
     node *next;
3;
typedel struct node *NODE
HODE getHode() {
     NODE temp;
     temp = (HODE) malloc (size of (struct node));
     if (temp == NULL) {
           paint (" Memory full (n');
           return NULL;
   neturn temp;
void freeHode (HODE temp) {
       free (temp);
3
HODE insert Front (HODE first) {
         NODE temp;
         temp = getrodec);
         int num;
        sconf(" ".d ", & num);
        temp -> info = num;
        temp - ) next = NULL ;
        if (first = = HULL) {
          return temp;
```

```
temp - next = fiast;
      fiast = temp .
      getian light;
HODE insert Pos ( NODE first, int pos) {
       int num;
       scan( " xd ", & num);
       NODE temp, cute, prev;
       temp = get Hode c);
       temp = value = run :
        temp -> next = NULL;
       Achest = NULL) {
                                    retion temp;
       int count = 1 :
        if (pos == count) {
              temp => next = first;
              Just = temp;
             netwn first;
       3
      cula = jiast;
       prev = NULL;
       while (cuar 1 = NULL) {
           if (count == pos) {
               bacak;
           Prev = curr;
           cush = cuso -> next;
           count++;
```

3

```
if (coor = = NULL) {
          paintf(" Cannot insert In");
          action first;
   temp - next = pher - rext;
   prev - next = temp;
   netwon first;
3
       insertRear (NODE frast) {
          NODE CURR, temp;
          int nun;
          temp = get rlode();
          scanf("1.d", & num);
          temp - value = num;
          temp-s next = NULL;
         Carif (first = = HULL) &
                actuan temp;
         cuan = first;
         while (cula -) next! = xlull) {
              cula = cula - next;
        cula - mext = temp :
        getwen first;
NODE delete Front (NODE birst) {
       NODE temp;
       if (first == HULL) {
            print( List is empty In");
           neturn likst;
```

```
12(19st -> next = = HULL) {
       painty (" Deleted element = % od |n", first -> value);
       preckode (jisst);
        aetian HULL;
  was temp = first;
    temp = temp ->next;
     print(" Deleted element = "1.d In", first-value);
     precklode (jinst);
     neturn temp;
NODE deleteReal (NODE just) {
      MODE cult;
      NODE PREV ;
      if (first = = MULL) {
           print(" List is emptyln");
         action first;
      if (first -) next == NULL) {
           paint (" old In", frast - value);
          jaceHode (jiast);
          neturn NULL;
     cusa = fisst;
      PREV - HULL;
     while (cusa -) next ! = HULL) {
          pher = cuan;
           cula = cum - next;
```

```
paer whexts HOLL;
    pant (" hdn", casa - value);
    przeriode (casa);
    Action filst;
HODE delete Pos (HODE fixst) int pos) {
         NODE cula, paer
          int count=1;
          if ( first = = HULL) {
              paraty (" rist o empty in ");
              action first;
         if ( prastre count == pos) {
               painty (" ord", first-svalue);
               cula = first;
               frast = fiast - next;
              freeHode (CLAO);
         char = first;
         paer = HULL;
         while (cula ! = NULL) {
             if (count = = pos) {
                 bacak:
             paer = cura;
            ours = curr - next; count ++;
       ight count = En
       if (ende == NULL) {
            paints ("invalid In"); actuan first;
```

```
paer -> link = cuan -> link
  peerlode (cusa);
  action flast;
void display (HODE first) ?
       HODE curse = first;
       if (culor = = NULL) {
           Pf("list is empty In");
          netion cula;
      while (cusor) = HULL) {
           printil "d.d", curs - value);
          Cula z cula -> next;
    print("In");
int main () {
     int ch; NODE just = HULL;
     int pos;
     while(1) {
        point ("enter In 1-inscription) n2-inscription In3-inscriptos In
               4 - deleteRear In 5 - deleteFront In 6 - deletePos In
               7 - display In 8 - exit In ");
       Scan (" olod", &ch);
      Switch (ch) {
```

```
11915 = inscrit Front (1915);
    break;
case 2:
    filest = isouthead (jest);
     break;
cose 3: scanj(".1.d", & pos);
     frast = inseat PosCfiast B, , pos);
     break;
case 4:
     first = delete Front (first);
     bacak;
 case 5:
      fiast = delete Rear (fiast);
     break;
case 6:
      scanf (" 1.d", & pos);
      juast = deletePost ( jiast, pos);
      baeak;
case 7:
      display ( first);
      break;
case8:
      netwon o;
```

case 1: