```
36) Circular Quene
#include < stdio. h>
#include (stdlib.h>
H define Size 50
int assign ];
 int sear = -1;
 int pront = - 1;
int Isfall ()
  if ( front = = ( seas +1) .1. Size)
     ; O newber
   else
     neturn -1;
   int IsEmpty()
    y(pont==-1 && rea ==-1)
      g return 0;
       che
        return -1;
    void Enqueu (int x)
        int item;
        if ( Is Full ( ) == 0)
           print ("Queue ouezon \n");
            return;
           rout = 0;
        else
         sua = ( sua +1) 1/. Size;
       Q[ seen ] = 20 3
```

```
int Dequeue ()
      printf("Queue underflow \n");
else
        ig ( pont == seas)
          x=Q[pout ];
front=-1;
oua;=-1;
         X = Q[ prond ];
         front = 1 ( growt + 1) 1. Sig;
        suturn x;
 Void Dieplay ()
    y( IsEmpty () == 0)
       prints ("Queue is emply \n');
        print { ("Queue clement : \n');
        for(i=front; i!=sea; i=(i+1):/. Size)
           ([:110, "n/b.v.") Abise q
         print (" . . d In', Q[:]);
void main ()
    ind choice, x, b;
   their while (1)
```

```
prints ("! Enqueue, 2. Dequeue, 3. Dieplay, 4. Exit In");
 print ("Enter your choice: In");
  Scanf (" 1. d", & choice),
  Switch (choice)
      print [l'Enter the number to be inserted into the queue's)
    case 1:
      san [(" /.d", &:x);
      trquem (x),
                                    一、(1) 加州外一大亚
       break;
                                 With the hard the
     Case 2:
        b= Dequeux ();
         print & (" " I'd was removed from the queue \n", $ b);
         break;
     Case 3:
Display();
break;
       CoxH: exit(1);
            print[["Invalid input \");
output !
 1. Engleu , 2. Dequer, 3. Display, h- Exit
 Enter the number to be enterted into the queue: 23
  1. Engues, 2. Deques, 3. Display, 4. Exit
 17600
    entre your choice:
   enter the number to be inserted into the queene.
  1. Eugen, 2. Degren, 3. Dieplay, h. Exit
    end your choice:
    23 was removed from the queue
     1. Engue, 2. Doguem, 3. Dixplay, 4. Exit
      entre your choice:
                                  That I you
      queue elements:
```

```
Single Linked list
   #include < stidio. h>
   # include < stalib. h>
      struct node
          int date;
         struct node * next;
        roid display ( );
        void insert-begin();
         wid insert and ();
        · Void insul- pos(2)
        Struct node + head = NULL;
        reid display()
          print] (" Elements are: 199;
          strud node * pt;
           ig (head == NULL)
              printf(" List is empty In");
            y sutwin;
             pts - head;
             while (pt.) = NOLID To a ways of
              print[ ("/.d | n", pts -> date );
             ptipti >next;
void invert-begin ()
  stuct node * temp;
   temp = (struct node*) mallo c (size of (struct node));
   printf(" Enter the value to be inserted \n");
    scan [ ("1.d", & temp->data);
    temp -> next = NULL; It was because
     y ( head = = NULL)
        head = temp;
     gelik
     temp -> next = head;
y head = tent;
```

```
noid meest-end ()
  skuct node + temp, + pts;
   demp = (Struct mode *) malloc (rizg (stand mode));
   printy ("Enter the value to be inserted in");
   scan (".1.d", & semp -> data);
    temp -> next = NULL;
    if ( head == NULL)
     dre
       pt = head;
        rolile (ptr->nent != NULL)
           i txen = ptg = ptg
         pte -> next = temp;
  uoid insert-posse()
    int pos, i;
    skut nade * temp, * pti;
      point ("Enter the position");
     san ( 1.d", & pos);
      temp=(Struct node *) mallor (Rize oz (stend node));
      print ("enter the value to be insufed ("");
       Kan & (".1.d", & temp -> data);
       temp-> new + = NULL;
        in ( bor == 0)
        ' temp -> next = head;
head = temp;
      eln
        por(i=0, pte=head; ikp8-1; i++)
            pt = ptr -> next;
      femp->next = pt ->next;
pt->next = temp;
```

```
roid main ()
 int choice;
 (1) shide
  print ("In1. to insect at the
            2. to insect at the end
             3. to insert at the polition by
             4. to display In 5. exit In").
     mints ("Enty your choice: In");
      san ("/. d", &choice);
       Switch (choic)
         Case 1:
             insulbegin();
             break
          lax 2:
              insert - end ();
              break;
           lasis!
               insut_pos();
               break;
            Cary:
                display (1)
                break;
             Coses:
                 exiteo);
                 break;
             dejoult:
                 printy ("Invalid
                 brook;
```

```
0/19
  1. to invot at the beginning
  2. to insert at the and
  3. to insent at the position
  4. to display
  s, exit
  enter your choice:
  enter the value to be inserted
   12
  1. to insert at the beginning 2. to insert at the end of 3. to insert at the position 4. to sliplay
   5. exit
  enter your choice
  enter the value to be inserted
  13
  1. to insert at the beginning 2. to insert at the end
  3. to intest at the position
   4. to inent at duplay
   5. exit
  enter your choice
   clineach are!
   12
   1. to insect at the beginning 2. to insect at the and
   3. to input at the porition
   4. to display
  5. exit
  enter your choice:
  enter the of value to be inserted
  8.0
  to to intest at the beginning
  2. to insert at the end
3. to insert at the position
4. to display
5. 0...
  s. frit your choice:
  enter the position 1
  enter the value to be inverted
   999
  elements are:
   12
   13
    80
    1299
```