```
Programu ! - 7
WAP to implement Single Linked list with the following operations:
(a) sort the linked lit
                                THOSE STATE TOWN BURNEY
(b) Reverse the linked lut
                               STREET PROBE CLASSICE - A MICE
       concatenation of two Unked dists
 (C)
 (d) implement - Stack and Quene using hinked list Representation.
                               KANKS LIGHTLES TO INTO
                            249 to be son the lunger of
(a) Sorting of Unked Ust
- Algorithm
                               France - Lover
a) define a node Curred which will point to head
 b) define another mode index which will point to mode next to current
(c) compase data of current and Index node . It currents data is
     greater than Index's dotathen, swap the data blo Them.
(d) Current will point current, next and index will point to
                     peters of trad luthory a rest of
   inder, next
 Q. Continue this process until the centire listis somed.
                      NOW ( Hange Mest to comount
 it void sortust (self) {
      · current = selfshead;
                          The 15 while actions save
      it ( self -> head == None) }
                 return)
        alsc &
            while (Current 1 = None); &
              index = current -> next;
          www.le(index!= Node){
           1 & Current => dasa > indexidates
              index. data = stemp.
                   fem P = Current. data;
                   current, dala = index.dafa;
                     index.dota - temp;
                     index = index. next;
                    Current = Current-next;
```

```
D> Reversing the unked list
    void Reverse & smid Nodo* head-red)
                                           +23 bright at
                 Node + Prev = Null;
           Struct Node* curred = * head = ref;
             Struct hode & next = null;
              while ( Current 1 = NULL) of bons businesses
                  next = current -> next;
                   current -> next = Prev;
                                            grand of linked list
                   prev = current;
                      current = next;
                  define a node word which will point to heady
                  * neadored = prev;
    define another mode index robids at the modernest to come
 Algorithm. 1. Initialize Those points prevas NULL, Curras head
         and next as Nuc.
         2. I terate through the lined Ust - In loop, do following
                    Europe . next and inde
             Before changing next to current, store next node.
      b (4) next = curr >next
                                                  were, against
              Now charge next to current
          (6) This is when actual reversing happens to
                                       Current = suffmant;
                 Current -> next = preu
                                         LOGEN = NOON
             (9) Move prevand current one step forward.
                                          = post - 1 105 1
                        Prev = Cur :
                                       · nestar
                        cum = next;
                             happine (Cumped ! = None); &
                            Charle - manin = what
                                  were (index 1= Node) &
                         Cotobyabai & some + son 1) } +
                                indees, have = steps.
                        Castert data)
                      allah. notini = mush, warrang
                            ignation who report
                         index - index nexts
                      Gunest = Texamos, nexts
```

```
- Proid Concatenation of two Linked list:
    void Concatenate ( Stoud node * a, Stouch node *b)
                        Morlos ( trior, 1886 ( 14 mp))
     E
          If (a->next == NULL)
             a -> next = b; [ Shah = last
          concatenate ( a snort ib);
    3
    void Concatenate (struct node * a, strudnode * b) ?
          , & Cal = MATE SE PI = MATE dest extend to see
                 it (a-) next == NULL) mit tomaste
                                     (dust: ) 1,
                      a -> next=b;
               else concaterete carsenext, b) juil
                 Printf ("Eina dor b is Nullin");
Di Implementation of Queneard Stack in Linked Ust:
                         Front is - West 1 02 i = word
 Stack:
+ ison > post() roop of an si wow i lideon
- 108 and - trouble
                        To add to seem of me on read dieny
+> push()
                                       vad adde (i) iteal
 A POPC)
                     if ( don't at least , don't some )
       Hodefino Max 10
                            temp a dolo = item;
        typedef. Stouct &
                Stouch & illustration of the quite
       type def struct stack * Stackporter type oct (truct &
         type of Struct &
                 element dota;
stack pointer unk;
                                 ebe .
                  3 Stack; The state of
              Stack pointer top [Max];
           top [i] = NULL , O & i < Max - Decks.
```

```
Void posts (inti, element item)
      Stack pointer tempjete to topbor haste ) There was not being
      Malloc (temp, riger (#temp));
         temp -) data = Item;
           temp sint = toptil
            top [i] = temp;
 void pop (int 1)
              world concernate (stores makes a stores ) lines
    Stack pointer temp = top[]
        element item; (sans
         it (! temp)
           return stack Empty ();
            Item = tem sdata soust as me
             topcij = + em = line;
             return Hem;
             fee (temp);
         rg.
   Turplem metrion of Queend Stacken Linked Wat :-
queue,
    front [ ] = NULL , OS i < Max
      Front[i] = NULL It he im queue's empty.
  10 add to reat of me unked queue
  void adda (1, 1+cm)
       queue pointer temp;
      malloc (temp, Size of ( *temp));
                               da xon andabit
         temp > data = item;
           temp + Link = NULLY
            it (front [i])
                 rear [i] - Line = temp;
                                Principles you
             else.
                Front [i) = tenp;
            rear [i] = temp.
```

to any among income was fill hope

```
void deleteq (mti) { /* Delete Pom the front of a linked quent

quenepointor temp = front [i];

element item;

It (!temp)

return queue Empty();

item = temp -> data;

front [i] = temp -> link;

free (temp);

return item;

3.
```