```
13) MAP to implement Queues using strigle contest using
                                                               neurode -> data: neurodata;
                                                               neurode -> nert = NULL;
# include < stdio.h>
# include <skelib.h>
                                                               it ( is Empty (Queue)) }
                                                                  oueue -> front : Queue -> 21ea21 : new Node;
Stouct node }
 int data;
                                                                 else 3
 storuct node * next;
                                                                   Queue -> greage -> next= newNode,
                                                                   Queue - greage : new Node;
                                                                 parintf ( "Ad enaucued to the Queue In", new Data);
 Stouct Querie 3
   staud node * front;
   stouch Node+ steast;
                                                                noid decurere (stouct aneuex ancue) &
 3;
                                                                     If ( "SEMPTY (Queue))
 Stouct Queue + initialize Queue () }
                                                                       posintf(" oneve is empty. cannot decureue \n");
    Stomet Luene * queue: (Stomet (Luene *) malloc (Sizeot (
                                                                       oretusin;
      Stouct Queue));
   neturn oucue;
                                                                      Stouct nocle * temp: Queue -> hont;
                                                                     Queue -> front = temp -> next;
                                                                    It (ouche -> front == NULL)
int isEmpty (stouct Queue * Queue) }
     Stetusin (aneue -> front == NULL);
                                                                       Queue -> gieagi = NULL;
void enqueue (sistuct aucue + queue, int neur Data)
                                                                   possible (" a decrued from lineed list ducue") temp-> docta);
                                                                   free (temp);
  Stouct node * newNode : (Stoud Node *) malloc ( $13006)
  (Stouct nocle);
```

```
moist boujulemene comment
                                                                Eccali output
                                                                 3 enquedued to the ducue
  if ( is Empty (aveue)) }
                                                                 4 enqueued to the oucue
    posintf ("Queue is empty". In");
                                                                   enamened to the anene
     neturn;
                                                                 3 dequeved from the Queue
                                                                 & 4 decureued from the dueue
  posintf (" Queue Elements");
  Stoud node * current = Queue - front;
                                                                     dequenced from the Quene
 while ( current != NULL)
                                                                Queue is empty
                                                               14) WAP to implement Doubly linked dist, insent a new node to left,
   posintf (" /d", wrient -> data);
                                                                  delete node based on value.
    current = current -> neach;
                                                               # indude <st dio.n>
                                                               # include < stalib. h>
  porinte ("(n");
                                                               sisuret node?
int main()
                                                                  int Data;
                                                                  Stouct node + prieu;
  Stouct Queue + queue : initialize Queue();
                                                                  Stouct node * next;
                                                                 3;
   encueve ( aureres oneve, 3).
                                                                stouct node + crecute node (int double) }
  enaucue (ducue, 4);
                                                                 stanct node * neunode : (Stanct node *) malloc (size of C
  enaucue (dueue, s);
                                                                 Stourst node));
  pointaueux (aueux);
                                                                 newnode -> data = data;
                                                                 newnode -> poreu: NULL;
  000 004E000
  deanene (anene);
                                                                 newhode -> neout = NULL;
  deavene (anene);
                                                                return new nocle;
 dequeue (queue);
 parintanene (anene);
                                                               word insertlest (stouct node ** head, Stouct Node * target, int date)
 greturno;
                                                                Stauct node + new Node = createrode (data);
```

```
c it (target -> poseu != NULL)
                                                                   current = current -> next;
    thought experience consider
   target -> poleu -> next : new Nocle ;
                                                                   pount (" Node not found \n");
   newNode -> poleu: target -> poleu;
                                                                   word display hist (Stomet Nocle * head)
 newNode -> next: target;
 target -> poreu -> next = new Node;
                                                                      posintf (" Doubly linked list");
 newNode -> porev : target -> porev;
                                                                      while ( head! : NULL)
                                                                        pountf(" /.d , head > data);
newNocle -> next : target;
                                                                        head; head->next;
 target -> porev = neurode;
 if (target == * head))
                                                                      CONTRACTOR OF PROPERTY
     * head == new Node;
                                                                    "m moun () }
 world deletenade (stouct node +* head, intualle)
                                                                    Stouct Node * head = NULL',
 Hower Node * current : * head;
                                                                    int Nodes wunt;
 while ( wrient! = NUL)
                                                                     pountf (" Enless the numbers of nocles");
                                                                    scanf ("1.d", & nodes cound);
    if ( whent -> data == uame)
                                                                    for Cinti: 0; i < nodes count it) }
     if ( current > prieu! = NULL)
                                                                     Put data;
        current -> poleu-> nevet : current -> nevet;
                                                                     PSTIPHT (" ENGOST CLOWD for the nocke (44 /.d, 1+1))
                                                                    From t Nocto* new Node + create Nocle (data),
       if ( current -> next! = NULL) }
                                                                    310 met Node * nem Node : create Norde (data);
         arrent -> nonu > porcu : arrent -> porcy;
                                                                     if ( head :: NULL)
    of ( wryent == * head)
                                                                         head; newNode;
            * head = current -> noxt;
                                                                         dsel insertlest (& head, head, data), 33
                                     J free (current);
```

,	
pointf (" Enter hame to be deleted"))	26/02/24
scanf (", a", ", " ualuetodelete);	* Pologoiam to Delete nocle in a BST
display list chead);	
o1ctuo1n0;	Stand Tarechode * manuallerode (Stanct Tarechode nod
5	Stouct Towerode* current = nocle;
Sample Input output	white (current & aurrent -> left != NULL)
Enter the number of nodes for the doubly dinited dist = 3.	current = current -> left;
Enter me nature for the doubly dinked list:	ocetuoin current,
	3
2	
3 class	stand tarcenode * deletenode (stand Tarcenode * 2000),
	int (ccy)
Double Linked List: 3 2 1 NULL	3
Enters the value to be deleted from the list 3	if (ROOT = NULL) Stetuan 3100t)
2 1 NULL	if (Key < 2100t -> ual)
	oroot > left = delete Node (5100+ -> left, key))
	electif (key > 9100t-> ual)
	. stoot > stig'ut = deterencele (stoot > stight, keg)
	el3C }
	if (stoot -> left == NULL)
	3

· late Cotten Constitution . Constitutions

, Display List Chead)

```
Void inordest te une esal (stouct Toceenode * 9100t) }
19/2/24 - 0001 aven aller - oftenschap
 Binary seasien toice: Toicherse using pricordes, postorder
                                                                    if ( groot ! = NULL) ?
  I morden , display the torce
                                                                        ?nordesitsial (sioot -> left);
                                                                        pointf ("1'd", stoot ->data);
                                                                        inordenterauenzal (2000 +> sight)
Hin ande (staio.h)
# include (Stalib.h)
Stouct Toreenode?
      inf data;
                                                                void percordentenucersal (storuct Tercenode* 2001)
      Stouct Torcenode + left;
                                                                         if ( 9100+ != NULL)
      Stouct Towerode " Right;
                                                                           pountf (". d", groot -> data);
                                                                           percordentenewessed (groot >1eft)
Stouct Torcenade * insport (Stouct Torcenade * 9100t, int classe)
                                                                           poreographoromeouscul (snoot -> sight)
       if (9100+ == NULL)
         retugin (reate Node (data);
                                                                word postobideontonomeonsal (Stouct Toneenode * oroot)
        if (data < 9100+ -> data)
             9100+ -> left = Priscout (9100+ ) left, data);
                                                                       if ( Droot ! = NULL)
        else if Edala > 2000t -> dala)
                                                                          post translem tenurement assert assertance and the post
                                                                                                  (9100t -> left)
               groot -> sight = inscort (groot -> sight, dura),
                                                                           postordesitsianesisal ( 91001 -> sight );
               neturin root;
                                                                          porintf ("V.d", groot->data);
```

```
void display ( stouct and Torcenode * 0,00+)
      polintf ("Inordes tolamesisal:");
      inordentorancesisal (2000);
       porint ( "\n");
      pount ( " porcordeon toraneonsal: ");
       percordeenterancersal (eroot);
       polintf ("\n");
       porintf ("postorider toraneousal");
       postordes terancoisal (erout);
       porinte ( a (n11);
 int main ()
        Stauct 1 sieenode * stoot = NULL;
         Proof = inscort ( Proof, So)
         insegut (9100+, 30);
         insport (900t, 20),
         insegut ( 9,001,40);
         insegut (9100+, 70)
         inscort (9100+,60);
         insegut (groot, 80);
         display (200+);
```

lect code psublem: Find bottom left torce 40fel (8802+1010) void find bottom left (stouct to eenode + node, int depth, int * moodepth, int * leftmostuallie) of chode == NULL) 3 gretusin; if (depth > + monodepth) & * maxdeptn = depth; * lejtmostualle = node-) ual; find bottomleft (node -) left ideptuil, mascaleptu, (estmostualle), find bottom left (node -> ought, deptuti, max Deptu, lest most nature). int findbottomualle (stauct toree node & coulte) int mascdophi= 0 ; int leftmostualle = 5000+>ual; Jand Bottomleft (2000), 1, Lmascdepty, & leftmost neturin leftmostualie; 9