```
34.63.30/
Breadth First Search
Hinclude < Stdions
# include & Stalib.h>
 Struct queue s
  int otems [SEZE];
  int front;
   int rear;
  Struct node {
       int vertex;
        Struct mode* next;
   struct Graph &
        int numVertices;
        truct node * * adjlists.
        int & visited;
  void bts ( struct Graph* graph, int startvertex)
      Struct queue* 9 = createqueuell;
       of raph -> visited [start vertex)=1;
       enqueue (a, startwertex);
```

```
white (! is Empty (9)).
      printaueue (9);
      înt currentvertex = dequeue (q):
      printf (" visited 1. d 1 n", curdient very
     Struct note * temp= graph => andi List
     while (temp) {
         int adivertex = temp-> vertex;
         if (graph-> visited [adivertex] == 0]
             graph-visited [adj vertex]=1;
             enqueue (a,adivertex);
            temp-temp-7 nent;
void addedge ( Etruct Graph & graph, int sic
                         int dest).
       struct node * new node = create Node
                                       (dest)
       new node-7 nent = graph -> adi Lists[sic
        graph->adj'Lists[dest)= newnode
```

```
Enqueue ( struct queue * q, înt value)
Void
 1
        (9-7 may = = fize-1)
      1
         Printf ("Queue is full in")
      1
      else f
         1 (q-7 front == -1).
2
q-7 front=0;
          9-7 reag ++;
          9 > i°tems[9->reag] = value;
      deque ( struct queue *a).
   3
       int item;
       if ( is Empty (9)).
         printf (" Queue is Empty"):
        else
           item = 9+7 items [q-7 front);
       , 4-7 front ++;
           if ( a -> front > a - rear)
              printf ("Resetting queue "):
```

9-> frunt = 9-> reag = -1/ netwin item; output:-Adjacency list of vestex o 1-71-7 Adjacency list of vertex 1. 1-> 0-7. Adjacency list of rester 2: 3-71-70-7. Adiacency list of restex 3 2 - 7. visited 2 visited 3 vibiled 1 visited o.

DIS Hinclude & fldio h> Hinclude Laldlibbs ftruct mode { int vertex; Struct mode \* ment, y . Struct Graph & int numventices int \* visited: Struct mode \*\* adjlists: 4; void DES ( Struct Graph \* graph, int vertex) struct node \* adilist = graph -> adilists (vertex) Struct node \* remp = aditiet; graph-> visited[vertex]= 1; printf (" Willited v.d in ", Nertex); while (temp!= NULL) connected vertex = temp -> vertex; it (graph > visited (connected Werter) - - 0) DES ( graph, connected vertex): temp=temp->nent;

addedge ( 81 ruct Graph\* graph, int stop int word 9 etruct node \* new Node = create Node loes newNore -> ment = graph -> adilists [src). graph -> adilists[src]=newNote; new Mode = create Mode (8 rc) s newNode -> next = graph -> add Lists [src]. graph -> adilists [dest] = new Node: print Graph ( struct Graph\* graph) Dion 8 for (v=0; V L graph -> numbertices: v+p) struct node \* temp = graph -> adjust printf ("In Adjacency lists of vertex while (temp) printf (4 y, d => " temp => vertex); temp: temp=>next; prints("In");

output: Adjacency list of vertex o. 2-71-7. Adjacency list of vertex 1. 2-70-7 Adjacency list of vertex 2 3-7 1-70-7 Adjacency list of vertex 3 2-7 visited 2 Wisited 3 visited 1 wisted o.