```
26/2/24
 Week -9
 B&FS >> Bougadth Proust Second.
# Include estatio.h>
# Produde < stdleb. h>
  define SIZE 40
  stauct queue 2
      Int otems (STORE);
       set fount;
        int seeas;
  struct queue + count queue();
   void exqueue (stouct queue + q, bit);
   Int dequeue (struct queue *9);
   vold display (stouct queue og);
        selmpty (stouct queue *q);
       pointqueux (stout queux * 9)
   staut node of
         ent vertex;
         struct node *next;
       3%
    struct node - collecte Node (Snt);
   . otherst graph {
         Int num Vertices;
         stouet gode ** odj Lists:
         int a restled;
       ٠
ز ر
```

```
void bts (struit goaph "graph, int stood vortex);
   stouct queue = q = consalequenc();
   gough -> visited [start vertex] = 1;
   enqueue (q, stant Verlez);
  while (! & Bompty (9)){
     point greve (9);
     Int current vertex = degree (g);
     point ("visited./d vo", current vertex),
     cloued node *femp = googh - adjlots [current borden]
    while (dump) }
        Ent adjuster = temp -> vertex;
      of (graph -> visited [adjunter] = =0){
         graph - vesited (adjuster) = 1;
          enqueux (9, adjuviter);
 doued rade * coreal made ( int v) }
  stanct node = newHode = malla (5:200 (stanct node));
   new Mode -> verter =N;
    newhode -> next = NULL;
     section new Mode;
Asunt Jacoph tweategraph (Introventices) (-
   start graph & graph = mallar (resurt (stouct graph));
    greaph -> new orthers = vertices;
```

```
gouph -> adj Lists = malla (vertices > street connect rate x);
 graph - visited = malloc (vertices & sixe of (8nt));
fort?
for (=0; 2< vertices; 24+)}
     geraph -adjlish (3) = NOLL;
     graph -> visited C:)=0>
  sectum goigh;
  vold addledge (Asunt Joseph & gouph, But soc, But dest)1
     Stauct node * new Node = careate Node (dest);
   rem Node -> rent = graph -> add Lerts (sou);
   gouph -> adj lists (sorc) = rew Node;
   numblode = coelecte Node (soce);
    new Nodes > next = graph -> and rests (dest).
     graph - adjlists (dest) = nurrode;
  stourt queux corate June ()
    stouet queue *9 = malloc (size of (stouet queue));
       9-> fount =1;
        9 -> sear = 1;
        section 9:
 Int BRompty ( stoud queue of ) {
     of (g) seem == +)
           seetum X
     else
         seetum o;
```

```
word englittle (stout queue =9, int value)
 d et (g→sucan == sIZE -1)
     point (" on June ?s Full [");
  else { et (q-, fount == -1)
         q plount =0;
      9 > flems [g → see ] = value,
the dequire (Arrich great "9) }
  Int Hems;
  et (esempty(g))
    pt ("queue & Rompy"))
    Hem =- 1:
  ) else {
    Etem = 97 Plems [ 9 slown ]; ( 1 1 gaze ) is Tolo
   q -> fount ff;
                 - ((にはつかな) 実もはある
   of (a - bronts a-secon) (seconds) while
    of ( resetting queen);
     q -> bront = q -> lean =1;
 setum stem;
```

```
val post greene (stouest queue = q){
     Set ?= q - fount;
   ef (esemptya)){
       pt ("gueue & empty");
  relse i pel 'in gume contains In");
      for (8= g-sount; ?29-souar +9; 9+4)}
          pt (" 48", gr Hens (i));
   But man () {
      Arrient graph * graph = coreate graph (6)
     add lage (quaph, 0,1);
     add Roge (goedth, 0,2);
      add Roge (gosaph.1,2);
      oddledge (goraph, 1,3);
      addlidge (geaph, 2,4);
      addledge (gouph, 3.4);
       bls (graph, 0);
        - setum o.
```

```
Queue Contains
  O sweething quee visite o.
Julie Contans
2 1 respect 2
que contains
 1 4 vistided 1
queue contains.
4 3 wited 4
queue contains
3 presetting queue ussited s.
              word there same a confession
        Designation Wingras Supplies they
                Carpet & John
```

```
# 80 dude < 1482 h>
# & clude < Adlib.h>
  staut node f
         got vertex;
        Staut node " nont;
  struct node * covade Node (Potv);
   stouct locaph &
          int num Vertices;
          Sht & usited;
        stount note = afflits;
    35
  rold DFS (Asunt quepts & graph, Sat vortex)
       Stourt node * and list = graph - and Lists (vertex);
       stauct node * temp = adglest;
      greaph - isseted Everter J=1;
        pt ("ussted 1d In"; western);
     whele (femp! = NUL) }
         Int connected Vester - temp - vertex;
             (graph - yested [comeded vortex)=20)
             DES Grouph, connected vertex);
       temp - timp - next)
```

```
rolg borget doods (2) suret doody in doods) &
       Both.
       ton (U=0; 16 golaph -meunvertices, V+4) 5
     street node + temp = goeaph -> adflish (V);
       ptU in Adjacency Lest of violer (d in ", v);
     while (temp)}
           pt(".1.d -> " , temp => vertex);
          temp = temp - next;
         pf (" mu);
  int main() {
    Struck Gerap ( geraph, 10,2);
    struct graph * graph = weat laugh (4),
     addredge (geraph, 0,1);
    add Adge (gouph, 10,2);
      add Edge (geeaph, 1,2),
     addedge (goeaph, 2,3);
      persont (seeast);
       DPS (geoph,2);
       Seetwon 0
```

Output: Asfaceacy List of vertex U 2-11-Adjacency List of vertex 1 2007 Alfacency List of vertex 2 3→1→0→ Adjacency lest of vortex 3. 27. 48sted 2 visited 3 visited 1 visited a