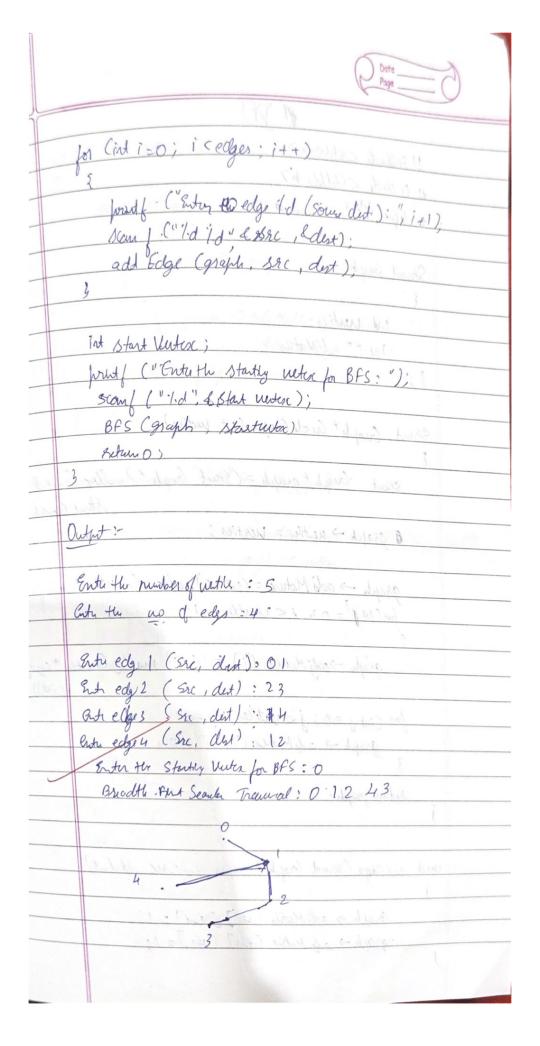


if (quew -> front == -1) grew -> stems Equeus -> rear ] = value, ist dequere (Street Queur \* queur) int Hem; if ( is Empty (quew)) print (" Queu is Empty"); else \* man 1 m 2) = man \* man a Stem = queu - items [queu -> front]; quem -> front = quem -> rear = -1; noton item; months. Street break ent wertices; 3; adj Matter; as - ward



_	and a set out of the
	Struct Graph * Create Graph (ent nection)
	Skurd Geraph * graph = (Stund Co. 115)
	Skut Graph * graph = (Stunt Graph *) mellor (Sizeof graph -> vertices; (Skut Graph
	graph > a di Matrix 2 (int xx)
	graph => adj Matrix = (int **) malloc (untiles * size of (int *));  for (int i =0; i= untiles; i++)
	graph -> adj Matrix [i] = (id *) malloc
	Curt malloc
	for (id j=0; j <vertice; j++)<="" td=""></vertice;>
	graph > adj Nation CiJ Ej] =0;
	setur geaph; 1 = 10 botton
	Canquer (quer. 1):
	The state of the s
	Void add Edge (Struct Grouph * graph, int suc, int dest)
	2 and a second and
	graph -> adjMatux (sec) [dut] =1;
	3 graph -> adj Matric ( dest J [serc] = 1;
	Word BCC Cours
	s Strat (right & graph, Int Start Weeks)
-	Int united [MAX_SIZE] = 203;
	Street Queux & queux = creat Que ();
	para se para cua (7)
	Visited @ [ Start Vertex ] = 1;
	Enqueue (quew, Start Kerted);
	print (" Breadth fires Dearch Tranges !");

	A 1.15 Hord & Hord Durie
	while (1 is Emply (quew))
	2 design (queu);
-	int curut Vertex = dequeu (queu);  pert (" "-d", current Vertex);
Chelgan 1	
	for (nt 120; i < graph > where; i++) &  prot f ("1.d", current Vertex);
111 0	Parent 1 ("1.d" current Vertix);
	for (inti=0; 1 < graph > neutron; i++).
((+	* 6 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4 4
	if (graph = adj Matrix [ curet Wester ] Ei] == 1
	Le visited (i]= =0)
	visiked Ci3=1; where mutes
	engrus (que, i);
	3
( )	The M of a Cara Cara hauf a contract in the E 1
	y
	fruit ("In") I raw I what who are dien
	3 11 (Co. 23 Chat co) retablished affects
	id mouln ()
	word PPS Carreit Graph " graph, But Shit Willeat? ?
	int vertices, edges, sec, dest;
	fruit ("Entry the newser of nextres!");
	scanf (1.0; & unter);
	Not the second of the board
	Strut Treagh + graph = Create Gregaph Chestras 73
	proself ("Enter the no of edges:");
	Scanf ("iled", Ledges);



	pr pfs
	# include <stdio. h=""></stdio.>
	weather cultiplishing
	# dellu MAX_SJZE @ 108
	1997 - 1897 - 1897 - 1897 - 19
	Skuit Graph
	-{
	ent curtics;
	int + * adj Matex;
	3; ( 2919 A man statement of ) but
	and the last of th
	Skut Graph & Grade Graph (Int mothers)
	skut Gregh + graph = (Skut Gregh *) rollog (seper)
	8 graph → nester = wester;
	g g special section of the section o
	graph -> adi Motux = (int * *) malloc ( untiles * siz f (in
	The tracker ( the state of the
	for (1st 1 = 0; 1 < rentles; 1+ +)
	for (1st ) = 0; i < vertles; 1+ +)
	for (1st   = 0; 1 < vertles; 1+ +)
	graph Fadj Matter (i) = (int *) maller (unte * syng
	graph = 2 adj Matur (Ci) = (int *) malla (unter * sym
	graph = 0; i < vertico; i++)  graph = 2adj Matur (Ci) = (int*) molla (unte * sym  (int))  for (int j = 0; j < verticos; j++)
	for (IN) = 0; i < vertico; i++)  graph = adj Mather (i) = (int*) maller (until * symplot (int))  for (int j = 0; j < vertices; j++)  graph = adj Mather (i) [j] = 0;
	for (IN) = 0; i < vertico; i++)  graph = adj Mather (i) = (int*) maller (until * symplot (int))  for (int j = 0; j < vertices; j++)  graph = adj Mather (i) [j] = 0;  3
	for (IN) = 0; i < vertico; i++)  graph = adj Mather (i) = (int*) maller (until * symplot (int))  for (int j = 0; j < vertices; j++)  graph = adj Mather (i) [j] = 0;
	for (IN) = 0; i < vertico; i++)  graph = 2 adj Mather (i) = (int*) mallor (unte * sym  (int))  for (int j = 0; j < verticos; j++)  graph = adj Mather (i) (j) = 0;  retur gaph;  }
U	for (IN) = 0; i < vertico; i++)  graph = 2 adj Mather (i) = (int*) mallor (unte * sym  (int))  for (int j = 0; j < verticos; j++)  graph = adj Mather (i) (j) = 0;  retur gaph;  }
U	graph = 0; i < vertin; i++)  graph = 2adj Matur (i) = (int*) mally (until * symples (int))  for (int j = 0; j < vertices; j++)  graph = adj Matur (i) (j) = 0;  return graph;  loid add Edge (Struct Graph & graph, int six, int dext)
U	graph = 0; i < vertin; i++)  graph = 2adj Matur (i) = (int*) mally (until * symples (int))  for (int j = 0; j < vertices; j++)  graph = adj Matur (i) (j) = 0;  return graph;  loid add Edge (Struct Graph & graph, int six, int dext)
	for (IN) = 0; i < vertico; i++)  graph = 2 adj Mather (i) = (int*) mallor (unte * sym  (int))  for (int j = 0; j < verticos; j++)  graph = adj Mather (i) (j) = 0;  retur gaph;  }



The owner of the last of	
	word DFS ( struct Graph "graph, ent spc, ind det)
	I
	goplead
	visited Estant What )=1;
	for (inti-0; i < graph > viertle; ?++)
	S Comment of the second of the
	( ( graph -) adj Mahler ( Start Wester ) Ci) == 1
	tevisité [i]==0)
	DFS (graph, i, visited);
	3 Le Maria Maria Maria Maria
	3
	il Ge convitta (Egedel)?
	ent is Connected (Stud Graph * graph)
	r continued
	int * wisted = (int *) mallor (graph -> weeker * sine d (int))
	int * visited = (int *) pralloc Graph -> weather * size of (int))
	for (IN i=0; i < graph > withles: 1++)
	for (it i=0; i < graph > wishles: 1++)  vished (i) =0;
	for (int i=0; i < graph > wishles: 1++)  visted (i) = 0;  DFS (graph, 0, visited);
	for (it i=0; i < graph > wishles: 1++)  vished (i) =0;
	for (it i=0; i < graph > untiles; i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i < graph > untiles; i++)  ?
	for (it i=0; i < graph > withles; i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i = graph > withles; i++)  if (visited (i) ==0)
	for (it i=0; i < graph > withlen; i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i = graph > withlen; i++)  ?  if (visited (i) ==0)  Netur 0;
	for (it i=0; i < graph > withlen; i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i= graph > withlen; i++)  if (visited (i) ==0)  return 0;
	for (it i=0; i < graph > withlen; i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i = graph > withlen; i++)  if (visited (i) = =0)  Neturn (i);
	for (IN i=0; i < graph > whiles; i++)  visted (i) = 0;  DFS (graph , 0, visited);  for (int i=0; i = graph > with ; i++)  if (visited (i) = =0)  letur 0;  return 1;
	for (it i=0; i < graph > while: i++)  visted (i) = 0;  DFS (graph, o, visited);  for (it i=0; i = graph > with: i++)  if (visited (i) = =0)  return;  return;  int mam()
	for (IN i=0; i < graph > wither; i++)  visted (i) =0;   DFS (graph , 0, visited);  for (int i=0; i = graph > wither; i++)  i ( (visited (i) = =0)  setum 0;  int make ()  int make ()
	for (IN i=0; i < graph > wither: 1++)  visted (i) = 0;  DFS (graph, o, visited);  for (int i=0; i= graph > wither: i++)  if (visited (i) ==0)  return;  int main()  int main()  int wither, edger, src, dest;
	for (IN i=0; i < graph > wither; i++)  visted (i) =0;   DFS (graph , 0, visited);  for (int i=0; i = graph > wither; i++)  i ( (visited (i) = =0)  setum 0;  int make ()  int make ()

Stut Greegel & graph = Create Graph (untiles 2) Many (" 1.8", Ledger); for Citi= 0) ic edger; i++) prid ( Entre edge 1 of (SAC dest): ", i+) scanf ("Y.d /d", LBR(, Edint);
add Edge (graph, Mc, dent)) if is comeated (Graph 1) fred ( o the graph is connected. In"); frul ( The graph is not cornected . In"); and the up of wenter 25 holy the no of edgs: 4 Entredge ( Src dest) = 01 Rtu Edg 2 ( Ste dest) = 23 Ety edg 3 ( Sec dest) - 33 4° Stu edge 11 C Sec dest) - 34 The graph 8 not Considered