

IT5412 ADS Lab Assignment (Graphs)

Swaminathan Navinashok - 2019115126

- 1. Write a C++ program to read a graph, display its adjacency matrix and sort its vertices using Topological Sort.**

```
#include<iostream>

using namespace std;

int n;

int adjacency[100][100];

void Graph();

int queue[100], front = -1,rear = -1;

void enqueue(int v);

int dequeue();

int isEmpty_queue();

int indegree(int v);

int main()
```

```
{
```

```
int i,j,v,count,topo_order[100],indeg[100];
```

```
Graph();
```

```
cout<<"\n\n\n\n\n\n\n\n\n\nThe adjacency matrix is \n\n\n\n\n\n\n\n";
```

```
cout<<"      ";
```

```
for(j=0; j<=n-1; j++)
```

```
cout<<" v"<<j<<" ";
```

```
cout<<"\n\n";
```

```
for(i=0; i<=n-1; i++)
```

```
{   cout<<"   v"<<i<<"   ";
```

```
    for(j=0; j<=n-1; j++)
```

```
        cout<<"   "<<adjacency[i][j]<<"   ";
```

```
    cout<<"\n\n";
```

```
}
```

```
for(i=0;i<n;i++)  
{  
    indeg[i] = indegree(i);  
  
    if( indeg[i] == 0 )  
        enqueue(i);  
}
```

```
count = 0;  
  
while( !isEmpty_queue( ) && count < n )  
{  
    v = dequeue();  
    topo_order[++count] = v;  
  
    for(i=0; i<n; i++)  
    {  
        if(adjacency[v][i] == 1)
```

```

adjacency[v][i] = 0;

indeg[i] = indeg[i]-1;

if(indeg[i] == 0)

    enqueue(i);

}

}

}

if( count < n )

{

    cout<<"\n    No topological ordering possible, graph contains cycle\n";

    exit(1);

}

cout<<"\n\n\n    Vertices in topological order are : \n\n\n";

for(i=1; i<=count; i++)

    cout<<"  v"<<topo_order[i]<<"  ";

cout<<"\n\n\n\n";

return 0;

```

```
}
```

```
void enqueue(int vertex)
```

```
{
```

```
    if (rear == 100-1)
```

```
        cout<<"\nQueue Overflow\n";
```

```
    else
```

```
{
```

```
    if (front == -1)
```

```
        front = 0;
```

```
        rear = rear+1;
```

```
        queue[rear] = vertex ;
```

```
}
```

```
}
```

```
int isEmpty_queue()
```

```
{
```

```
    if(front == -1 || front > rear )
```

```
        return 1;
```

```
    else
        return 0;
    }

int dequeue()
{
    int del_item;
    if (front == -1 || front > rear)
    {
        cout<<"\nQueue Underflow\n";
        exit(1);
    }
    else
    {
        del_item = queue[front];
        front = front+1;
        return del_item;
    }
}
```

```
int indegree(int v)

{
    int i,in_deg = 0;

    for(i=0; i<n; i++)

        if(adjacency[i][v] == 1)

            in_deg++;

    return in_deg;
}
```

```
void Graph()

{
    int i,max_edges,origin,destin;

    cout<<"\n      Enter number of vertices :  ";

    cin>>n;

    max_edges = n*(n-1);
```

```
for(i=1; i<=max_edges; i++)  
{  
    cout<<"\n\n\n\n  Enter edge "<<i<<" ";  
    cout<<"\n\n      enter origin vertex : ";cin>>origin;cout<<"\n\n  
enter destination vertex : ";cin>>destin;  
  
  
  
  
    if((origin == -1) && (destin == -1))  
        break;  
  
  
  
  
    if( origin >= n || destin >= n || origin<0 || destin<0)  
    {  
        cout<<"\nInvalid edge!\n";  
        i--;  
    }  
    else  
        adjacency[origin][destin] = 1;  
}  
}
```

C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe

enter destination vertex : 4

Enter edge 6

enter origin vertex : 3

enter destination vertex : 2

Enter edge 7

enter origin vertex : 2

enter destination vertex : 5

Enter edge 8

enter origin vertex : 5

enter destination vertex : 1

Enter edge 9

enter origin vertex : -1

enter destination vertex : -1

The adjacency matrix is



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe

The adjacency matrix is

	v0	v1	v2	v3	v4	v5
v0	0	0	0	0	1	0
v1	0	0	0	0	1	0
v2	1	0	0	0	0	1
v3	1	0	1	0	0	0
v4	0	0	0	0	0	0
v5	0	1	0	0	0	0

Vertices in topological order are :

v3 v2 v0 v5 v1 v4

Process exited after 33.4 seconds with return value 0
Press any key to continue . . . ■



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe

Enter number of vertices : 6

Enter edge 1

enter origin vertex : 1

enter destination vertex : 4

Enter edge 2

enter origin vertex : 2

enter destination vertex : 5

Enter edge 3

enter origin vertex : 3

enter destination vertex : 0

Enter edge 4

enter origin vertex : 2

enter destination vertex : 0

Enter edge 5

enter origin vertex : 0



Type here to search



```
█ Select C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe
      enter destination vertex : 1
```

```
Enter edge 4
```

```
      enter origin vertex : 1
```

```
      enter destination vertex : 4
```

```
Enter edge 5
```

```
      enter origin vertex : -1
```

```
      enter destination vertex : -1
```

```
The adjacency matrix is
```

	v0	v1	v2	v3	v4
v0	0	0	0	1	0
v1	0	0	0	0	1
v2	0	0	0	0	1
v3	0	1	0	0	0
v4	0	0	0	0	0

```
Vertices in topological order are :
```

```
v0      v2      v3      v1      v4
```



```
Type here to search
```



```
█ Select C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe
      enter origin vertex : -1
```

```
      enter destination vertex : -1
```

```
The adjacency matrix is
```

	v0	v1	v2	v3	v4
v0	0	0	0	1	0
v1	0	0	0	0	1
v2	0	0	0	0	1
v3	0	1	0	0	0
v4	0	0	0	0	0

```
Vertices in topological order are :
```

```
v0          v2          v3          v1          v4
```

```
-----  
Process exited after 28.35 seconds with return value 0  
Press any key to continue . . . ■
```



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q1\Project1.exe

Enter number of vertices : 5

Enter edge 1

enter origin vertex : 0

enter destination vertex : 3

Enter edge 2

enter origin vertex : 2

enter destination vertex : 4

Enter edge 3

enter origin vertex : 3

enter destination vertex : 1

Enter edge 4

enter origin vertex : 1

enter destination vertex : 4

Enter edge 5

enter origin vertex : -1



Type here to search



Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1 main.cpp

```
18 int main()
19 {
20     int i,j,v,count,topo_order[100],indeg[100];
21
22     Graph();
23
24
25     cout<<"\n\n\n\n      The adjacency matrix is \n\n\n\n ";
26     cout<<" ";
27     for(j=0; j<=n-1; j++)
28         cout<< " v"<<j<< " ";
29
30
31     cout<<"\n\n";
32     for(i=0; i<=n-1; i++)
33     {
34         cout<< " v"<<i<< " ";
35         for(j=0; j<=n-1; j++)
36             cout<< " "<<adjacency[i][j]<< " ";
37         cout<<"\n\n";
38
39         for(i=0;i<n;i++)
40         {
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1 main.cpp

```
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1

main.cpp

```
49 while( !isEmpty_queue( ) && count < n )
50 {
51     v = dequeue();
52     topo_order[++count] = v;
53     for(i=0; i<n; i++)
54     {
55         if(adjacency[v][i] == 1)
56         {
57             adjacency[v][i] = 0;
58             indeg[i] = indeg[i]-1;
59             if(indeg[i] == 0)
60                 enqueue(i);
61         }
62     }
63 }
64
65 if( count < n )
66 {
67     cout<<"\n      No topological ordering possible, graph contains cycle\n";
68     exit(1);
69 }
70 cout<<"\n\n      Vertices in topological order are : \n\n\n";
71 for(i=1; i<=count; i++)
72     cout<< "      v"<<topo_order[i]<< "      ";
73
74 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1

main.cpp

```
65 if( count < n )
66 {
67     cout<<"\n      No topological ordering possible, graph contains cycle\n";
68     exit(1);
69 }
70 cout<<"\n\n      Vertices in topological order are : \n\n\n";
71 for(i=1; i<=count; i++)
72     cout<< "      v"<<topo_order[i]<< "      ";
73
74 return 0;
75
76 void enqueue(int vertex)
77 {
78     if (rear == 100-1)
79         cout<<"\nQueue Overflow\n";
80     else
81     {
82         if (front == -1)
83             front = 0;
84         rear = rear+1;
85         queue[rear] = vertex ;
86     }
87 }
88 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1

main.cpp

```
77 void enqueue(int vertex)
78 {
79     if (rear == 100-1)
80         cout<<"\nQueue Overflow\n";
81     else
82     {
83         if (front == -1)
84             front = 0;
85         rear = rear+1;
86         queue[rear] = vertex ;
87     }
88 }
89
90 int isEmpty_queue()
91 {
92     if(front == -1 || front > rear )
93         return 1;
94     else
95         return 0;
96 }
97
98 int dequeue()
99 {
100    int del_item;
101 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Line: 8 Col: 20 Sel: 0 Lines: 151 Length: 3639 Insert Done parsing in 0.016 seconds

20:23 22-05-2021

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project1

main.cpp

```
91 int isEmpty_queue()
92 {
93     if(front == -1 || front > rear )
94         return 1;
95     else
96         return 0;
97 }
98
99 int dequeue()
100 {
101     int del_item;
102     if (front == -1 || front > rear)
103     {
104         cout<<"\nQueue Underflow\n";
105         exit(1);
106     }
107     else
108     {
109         del_item = queue[front];
110         front = front+1;
111         return del_item;
112     }
113 }
114 int indegree(int v)
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Line: 8 Col: 20 Sel: 0 Lines: 151 Length: 3639 Insert Done parsing in 0.016 seconds

20:23 22-05-2021

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp

```
112 L }
113
114 int indegree(int v)
115 {
116     int in_deg = 0;
117     for(i=0; i<n; i++)
118         if(adjacency[i][v] == 1)
119             in_deg++;
120     return in_deg;
121 }
122
123 void Graph()
124 {
125     int i,max_edges,origin,destin;
126
127     cout<<"\n      Enter number of vertices :   ";
128
129
130
131     cin>>n;
132     max_edges = n*(n-1);
133
134     for(i=1; i<=max_edges; i++)
135     {
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp

```
128
129
130
131
132     cin>>n;
133     max_edges = n*(n-1);
134
135     for(i=1; i<=max_edges; i++)
136     {
137         cout<<"\n\n\n      Enter edge "<<i<<" ";
138         cout<<"\n\n          enter origin vertex : ";cin>>origin;cout<<" \n\n";
139         cout<<"\n          enter destination vertex : ";
140         cin>>destin;
141
142         if((origin == -1) && (destin == -1))
143             break;
144
145         if(origin >= n || destin >= n || origin<0 || destin<0)
146         {
147             cout<<"\nInvalid edge!\n";
148             i--;
149         }
150     }
151 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.90279674530029 Mib
- Compilation Time: 0.25s

Type here to search

Project1 - [Project1.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

(globals)

main.cpp

```
1 #include<iostream>
2 using namespace std;
3
4
5
6
7 int n;
8 int adjacency[100][100];
9 void Graph();
10
11 int queue[100], front = -1, rear = -1;
12 void enqueue(int v);
13 int dequeue();
14 int isEmpty_queue();
15
16 int indegree(int v);
17
18 int main()
19 {
20     int i, j, v, count, topo_order[100], indeg[100];
21
22     Graph();
23
24 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\ql\Project1.exe
- Output Size: 1.9027967453029 MiB
- Compilation Time: 0.25s

Type here to search

Line: 8 Col: 20 Sel: 0 Lines: 151 Length: 3639 Insert Done parsing in 0.016 seconds

20:23 22-05-2021

2. Write a C++ program to read a graph, display its adjacency matrix and traverse the graph through Breadth First search and Depth First Search algorithm.

```
#include<iostream>

using namespace std;

#define initial 1

#define waiting 2

#define visited 3

int n;

int adjacency[100][100];

int state[100];

void Graph();

void Breadth();

void BFS(int v);

int queue[100], front=-1,rear=-1;

void enqueue(int vertex);

int dequeue();

int isEmpty_queue();
```



```
cout<<"\n\n";
for(i=0; i<=n-1; i++)
{
    cout<<"    v"<<i<<"    ";
    for(j=0; j<=n-1; j++)
        cout<<"    "<<adjacency[i][j]<<"    ";
    cout<<"\n\n";
}
```

```
Breadth();
```

```
Depth();
```

```
return 0;
```

```
}
```

```
void Breadth()
```

```
{
```

```
int c,v;

while(1){

    for(v=0; v<n; v++)

        state[v]=initial;

    cout<<"\n\n\n      Enter starting vertex for Breadth First Search(-1 to exit) :

    ";

    cin>>v;if(v==-1) break;

    BFS(v);

    for(v=0; v<n; v++)

        if(state[v] == initial)

            BFS(v);

    }

}
```

```
void BFS(int v)

{
    int i;

    enqueue(v);

    state[v]=waiting;

    cout<<" \n\n\n      ";

    while( !isEmpty_queue() )

    {
        v = dequeue( );

        cout<<"  v"<<v<<"  ";

        state[v] = visited;

        for(i=0; i<n; i++)

        {

            if( adjacency[v][i] == 1 && state[i] == initial)

            {

                enqueue(i);

                state[i] = waiting;
```

```
    }

}

cout<<"\n\n\n ";

}

void enqueue(int vertex)

{

if (rear == 100-1)

    cout<<"Queue Overflow\n";

else

{

if (front == -1)

    front = 0;

rear = rear+1;

queue[rear] = vertex ;

}

}
```

```
int isEmpty_queue()
{
    if(front == -1 || front > rear )
        return 1;
    else
        return 0;
}
```

```
int dequeue()
{
    int del_item;
    if (front == -1 || front > rear)
    {
        cout<<"\nQueue Underflow\n";
        exit(1);
    }
    del_item = queue[front];
    front = front+1;
```

```
return del_item;

}

void Graph()
{
    int i,max_edges,origin,destin;

    cout<<"\n      Enter number of vertices :   ";
    cin>>n;
    max_edges = n*(n-1);

    for(i=1; i<=max_edges; i++)
    {
        cout<<"\n\n\n\n  Enter edge "<<i<<" ";
        cout<<"  \n\n      enter origin vertex : ";cin>>origin;cout<<" \n\n
enter destination vertex : ";cin>>destin;
```

```
if((origin == -1) && (destin == -1))

    break;

if( origin >= n || destin >= n || origin<0 || destin<0)

{

    cout<<"\nInvalid edge!\n";

    i--;

}

else

    adjacency[origin][destin] = 1;

}

}

void Depth()

{

    int v;
```

```
while(1)

{
for(v=0; v<n; v++)
    state[v]=initial;

    cout<<"\n\n\n    Enter starting node for Depth First Search (-1) to
break :    ";
    cin>>v;if(v== -1) break;
    DFS(v);
    cout<<"\n";
}

void DFS(int v)
{
    int i;
```

```
push(v);

cout<<" \n\n\n    ";

while(!isEmpty_stack())

{

    v = pop();

    if(state[v]==initial)

    {

        cout<<"  v"<<v<<"  ";

        state[v]=visited;

    }

    for(i=n-1; i>=0; i--)

    {

        if(adjacency[v][i]==1 && state[i]==initial)

            push(i);

    }

    cout<<"\n\n\n";

}

}
```

```
void push(int v)

{
    if(top == (100-1))

    {
        cout<<"\nStack Overflow\n";

        return;
    }

    top=top+1;

    stack[top] = v;
}
```

```
int pop()

{
    int v;

    if(top == -1)

    {
        cout<<"\nStack Underflow\n";
    }
}
```

```
    exit(1);

}

else

{

    v = stack[top];

    top=top-1;

    return v;

}

}

int isEmpty_stack( )

{

    if(top == -1)

        return 1;

    else

        return 0;

}
```

C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe

```
Enter starting vertex for Breadth First Search(-1 to exit) :      1
```

v1 v0 v2

```
Enter starting vertex for Breadth First Search(-1 to exit) :      2
```

v2 v1 v0

```
Enter starting vertex for Breadth First Search(-1 to exit) :      3
```

v3

v0 v2 v1

```
Enter starting vertex for Breadth First Search(-1 to exit) :      0
```

v0 v2 v1

```
Enter starting vertex for Breadth First Search(-1 to exit) :      -1
```



C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe

Enter edge 5

enter origin vertex : 2

enter destination vertex : -1

Invalid edge!

Enter edge 5

enter origin vertex : -1

enter destination vertex : -1

The adjacency matrix is

	v0	v1	v2
v0	0	0	1
v1	1	0	1
v2	0	1	0

Enter starting vertex for Breadth First Search(-1 to exit) : 1

v1 v0 v2

Enter starting vertex for Breadth First Search(-1 to exit) : 2



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe

enter destination vertex : 2

Enter edge 2

enter origin vertex : 2

enter destination vertex : 1

Enter edge 3

enter origin vertex : 0

enter destination vertex : 2

Enter edge 4

enter origin vertex : 1

enter destination vertex : 0

Enter edge 5

enter origin vertex : 2

enter destination vertex : -1

Invalid edge!

Enter edge 5



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe

```
Enter number of vertices :      3
```

```
Enter edge 1
```

```
    enter origin vertex : 1
```

```
    enter destination vertex : 2
```

```
Enter edge 2
```

```
    enter origin vertex : 2
```

```
    enter destination vertex : 1
```

```
Enter edge 3
```

```
    enter origin vertex : 0
```

```
    enter destination vertex : 2
```

```
Enter edge 4
```

```
    enter origin vertex : 1
```

```
    enter destination vertex : 0
```

```
Enter edge 5
```

```
    enter origin vertex : 2
```



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe

Enter starting node for Depth First Search (-1) to break : 0

v0 v2 v1

Enter starting node for Depth First Search (-1) to break : 1

v1 v0 v2

Enter starting node for Depth First Search (-1) to break : 2

v2 v1 v0

Enter starting node for Depth First Search (-1) to break : 3

Enter starting node for Depth First Search (-1) to break : -1

Process exited after 66.48 seconds with return value 0
Press any key to continue . . .



Type here to search



Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

main.cpp

```
25 void Depth();
26 void DFS(int v);
27
28 int stack[100];
29 int top = -1;
30 void push(int v);
31 void pop();
32 int isEmpty_stack();
33
34 int main()
35 {
36     Graph();int i,j;
37
38     cout<<"\n\n      The adjacency matrix is \n\n ";
39     cout<<" ";
40     for( j=0; j<=n-1; j++)
41     cout<<"   "j<<" ";
42
43
44     cout<<"\n\n";
45     for(i=0; i<=n-1; i++)
46     {
47         cout<<"   ";
48         for(j=0; j<=n-1; j++)
49         cout<<"   "v<<i<<" ";
50     }
51 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

main.cpp

```
47     {
48         cout<<"   v<<i<<" ";
49         for(j=0; j<=n-1; j++)
50             cout<<"   "<<adjacency[i][j]<<" ";
51         cout<<"\n\n";
52     }
53
54     Breadth();
55     Depth();
56
57     return 0;
58 }
59
60 void Breadth()
61 {
62     int c,v;
63
64
65
66
67     while(1){
68         for(v=0; v<n; v++)
69             state[v]=initial;
70         cout<<"\n\n      Enter starting vertex for Breadth First Search(-1 to exit) : ";
71     }
72 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
67 while(1){  
68     for(v=0; v<n; v++)  
69         state[v]=initial;  
70     cout<<"\n\n\n" Enter starting vertex for Breadth First Search(-1 to exit) : " ;  
71     cin>>v;if(v== -1) break;  
72     BFS(v);  
73     for(v=0; v<n; v++)  
74         if(state[v] == initial)  
75             BFS(v);  
76     }  
77 }  
78  
80 void BFS(int v)  
81 {  
82     int i;  
83     enqueue(v);  
84     state[v]=waiting;  
85     cout<<"\n\n\n" ;  
86     while( !isEmpty_queue() )  
87     {  
88         v = dequeue( );  
89         cout<<" v<<<<" ;  
90         for(i=0; i<n; i++)  
91         {  
92             if( adjacency[v][i] == 1 && state[i] == initial)  
93             {  
94                 enqueue(i);  
95                 state[i] = waiting;  
96             }  
97         }  
98         cout<<"\n\n\n" ;  
99     }  
100 }  
101  
102 void enqueue(int vertex)  
103 {  
104     if (rear == 100-1)  
105         cout<<"Queue Overflow\n";  
106     else  
107     {  
108         if (front == -1)  
109             front = rear = 0;  
110         else  
111             rear++;  
112         adj[front][vertex] = 1;  
113     }  
114 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
88 {  
89     v = dequeue( );  
90     cout<<" v<<<<" ;  
91     state[v] = visited;  
92     for(i=0; i<n; i++)  
93     {  
94         if( adjacency[v][i] == 1 && state[i] == initial)  
95         {  
96             enqueue(i);  
97             state[i] = waiting;  
98         }  
99     }  
100 cout<<"\n\n\n" ;  
101  
102 void enqueue(int vertex)  
103 {  
104     if (rear == 100-1)  
105         cout<<"Queue Overflow\n";  
106     else  
107     {  
108         if (front == -1)  
109             front = rear = 0;  
110         else  
111             rear++;  
112         adj[front][vertex] = 1;  
113     }  
114 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
184 void enqueue(int vertex)
185 {
186     if (rear == 100-1)
187         cout<<"Queue Overflow\n";
188     else
189     {
190         if (front == -1)
191             front = 0;
192         rear = rear+1;
193         queue[rear] = vertex ;
194     }
195 }
196
197 int isEmpty_queue()
198 {
199     if(front == -1 || front > rear )
200         return 1;
201     else
202         return 0;
203 }
204
205 int dequeue()
206 {
207 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
125
126 int dequeue()
127 {
128     int del_item;
129     if (front == -1 || front > rear )
130     {
131         cout<<"\nQueue Underflow\n";
132         exit(1);
133     }
134
135     del_item = queue[front];
136     front = front+1;
137     return del_item;
138 }
139
140 void Graph()
141 {
142     int i,max_edges,origin,destin;
143
144     cout<<"\n      Enter number of vertices :      ";
145
146
147     cin>>n;
148 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2 main.cpp

```
149 max_edges = n*(n-1);
150
151 for(i=1; i<=max_edges; i++)
152 {
153     cout<<"\n\n\n      Enter edge "<<i<<" "
154     cout<<"\n\n          enter origin vertex : "cin>>origin;cout<<" \n\n      enter destination vert
155
156     if((origin == -1) && (destin == -1))
157         break;
158
159     if( origin >= n || destin >= n || origin<0 || destin<0)
160     {
161         cout<<"\nInvalid edge!\n";
162         i--;
163     }
164     else
165         adjacency[origin][destin] = 1;
166 }
167
168
169
170 void Depth()
171 {
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2 main.cpp

```
169
170
171 void Depth()
172 {
173     int v;
174     while(1)
175     {
176         for(v=0; v<n; v++)
177             state[v]=initial;
178
179
180         cout<<"\n\n\n      Enter starting node for Depth First Search (-1) to break :   ";
181         cin>>v;if(v== -1) break;
182         DFS(v);
183         cout<<"\n";
184     }
185 }
186
187
188 void DFS(int v)
189 {
190     int i;
191     push(v);
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
188 void DFS(int v)
189 {
190     int i;
191     push(v);
192     cout<<"\n\n\n";
193     while(!isEmpty_stack())
194     {
195         v = pop();
196         if(state[v]==initial)
197         {
198             cout<<" " << v << " ";
199             state[v]=visited;
200         }
201         for(i=n-1; i>=0; i--)
202         {
203             if(adjacency[v][i]==1 && state[i]==initial)
204                 push(i);
205         }
206     }
207     cout<<"\n\n\n";
208 }
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
207     cout<<"\n\n\n";
208 }
209
210
211
212 void push(int v)
213 {
214     if(top == (100-1))
215     {
216         cout<<"\nStack Overflow\n";
217         return;
218     }
219     top=top+1;
220     stack[top] = v;
221 }
222
223
224 int pop()
225 {
226     int v;
227     if(top == -1)
228     {
229         cout<<"\nStack Underflow\n";
230         exit(1);
231     }
232 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 MiB
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
224 int pop()
225 {
226     int v;
227     if(top == -1)
228     {
229         cout<<"\nStack Underflow\n";
230         exit(1);
231     }
232     else
233     {
234         v = stack[top];
235         top=top-1;
236         return v;
237     }
238 }
239
240 int isEmpty_stack()
241 {
242     if(top == -1)
243         return 1;
244     else
245         return 0;
246 }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

Project2 - [Project2.dev] - [Executing] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project2

main.cpp

```
1
2
3 #include<iostream>
4 using namespace std;
5
6
7 #define initial 1
8 #define waiting 2
9 #define visited 3
10
11 int n;
12 int adjacency[100][100];
13 int state[100];
14
15 void Graph();
16 void Breadth();
17 void BFS(int v);
18
19 int queue[100], front=-1,rear=-1;
20 void enqueue(int vertex);
21 int dequeue();
22 int isEmpty_queue();
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q2\Project2.exe
- Output Size: 1.90444278717041 Mib
- Compilation Time: 1.59s

Type here to search

3. Write a C++ program to read a graph, display its adjacency matrix and to find the shortest path from the source vertex to destination vertex using Dijkstra's Shortest Path algorithm.

```
#include<iostream>

using namespace std;

#define ZERO 0

#define ONE 1

#define NIL -1

void Path(int s, int v );

void Dijkstra( int s);

int temp( );

void graph();

int n;

int adjacency[100][100];

int ancestors[100];

int path[100];

int status[100];
```

```
int main()
{
    int s,v,i,j;

    graph();
    cout<<"\n\n    \n    The adjacency matrix is \n\n ";
    cout<<"      ";
    for( j=0; j<=n-1; j++)
        cout<<" v"<<j<<" ";
    cout<<"\n\n";
    for(i=0; i<=n-1; i++)
    {
        cout<<"    v"<<i<<"   ";
        for(j=0; j<=n-1; j++)
            cout<<"    "<<adjacency[i][j]<<"   ";
        cout<<"\n\n";
    }
}

while(1)
{
```

```

cout<<"\n\n\n\n\n\n\n\n\n\n\n\n"; Enter source vertex :   ";

cin>>s;

Dijkstra(s);

cout<<"\n\n\n\n\n\n"; Enter destination vertex(-1 to
quit):   ";

cin>>v;

cout<<"\n\n\n\n\n\n";

if(v == -1)

break;

if(v < 0 || v >= n )

cout<<"\n\n\n  This vertex does not exist\n";

else if(v == s)

cout<<"\n\n\n  Source and destination vertices are same\n";

else if( path[v] == 9999 )

cout<<"\n\n\n  There is no path from source to destination vertex\n";

else

Path(s,v);

}

```

```
    return 0;  
}  
  
void Dijkstra( int s)
```

```
{  
    int i,current;
```

```
    for(i=0; i<n; i++)
```

```
{  
    ancestors[i] = NIL;
```

```
    path[i] = 9999;  
    status[i] = ZERO;
```

```
}
```

```
path[s] = 0;
```

```
while(1)
```

```
{  
  
    current = temp( );  
  
    if( current == NIL )  
        return;  
  
    status[current] = ONE;  
  
    for(i=0; i<n; i++)  
    {  
  
        if ( adjacency[current][i] !=0 && status[i] == ZERO )  
            if( path[current] + adjacency[current][i] < path[i] )  
            {  
                ancestors[i] = current;  
                path[i] = path[current] + adjacency[current][i];  
            }  
    }  
}
```

```
    }

}

int temp( )

{
    int i;

    int min = 9999;

    int k = NIL;

    for(i=0;i<n;i++)

    {

        if(status[i] == ZERO && path[i] < min)

        {

            min = path[i];

            k = i;

        }

    }

    return k;
}
```

}

void Path(int s, int v)

{

int i,u;

int path[100];

int shortdist = 0;

int count = 0;

while(v != s)

{

count++;

path[count] = v;

u = ancestors[v];

shortdist += adjacency[u][v];

v = u;

}

```
count++;

path[count]=s;

cout<<"\n\n\n    Shortest Path is   :\n\n\n\n\n";

for(i=count; i>=1; i--)

    cout<<"  v"<<path[i]<<"  ";

cout<<"\n\n\n\n\n\n    Shortest distance is :      "<<shortdist<<"\n";

}
```

```
void graph()

{

int i,max_edges,origin,destin, wt;

cout<<"\n\n\n    Enter number of vertices   :   ";

cin>>n;

max_edges = n*(n-1);

for(i=1;i<=max_edges;i++)

{
```


C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter edge 4

enter origin vertex : 1

enter destination vertex : 0

Enter weight for this edge : 3

Enter edge 5

enter origin vertex : 1

enter destination vertex : 2

Enter weight for this edge : 3

Enter edge 6

enter origin vertex : 1

enter destination vertex : 3

Enter weight for this edge : 7

Enter edge 7

enter origin vertex : 1



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter edge 7

enter origin vertex : 1

enter destination vertex : 2

Enter weight for this edge : 4

Enter edge 8

enter origin vertex : 2

enter destination vertex : 1

Enter weight for this edge : 8

Enter edge 9

enter origin vertex : 2

enter destination vertex : 3

Enter weight for this edge : 1

Enter edge 10

enter origin vertex : 2



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter edge 10

enter origin vertex : 2

enter destination vertex : 4

Enter weight for this edge : 9

Invalid edge!

Enter edge 10

enter origin vertex : 3

enter destination vertex : 1

Enter weight for this edge : 9

Enter edge 11

enter origin vertex : 3

enter destination vertex : 0

Enter weight for this edge : 7

Enter edge 12

enter origin vertex : 3



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter edge 11

enter origin vertex : 3

enter destination vertex : 0

Enter weight for this edge : 7

Enter edge 12

enter origin vertex : 3

enter destination vertex : 4

Enter weight for this edge : 2

Invalid edge!

Enter edge 12

enter origin vertex : 3

enter destination vertex : 1

Enter weight for this edge : 7

The adjacency matrix is

v0 v1 v2 v3



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter edge 12

enter origin vertex : 3

enter destination vertex : 1

Enter weight for this edge : 7

The adjacency matrix is

	v0	v1	v2	v3
v0	0	2	4	8
v1	3	0	4	7
v2	0	8	0	1
v3	7	7	0	0

Enter source vertex : 2

Enter destination vertex(-1 to quit): 0



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

The adjacency matrix is

	v0	v1	v2	v3
v0	0	2	4	8
v1	3	0	4	7
v2	0	8	0	1
v3	7	7	0	0

Enter source vertex : 2

Enter destination vertex(-1 to quit): 0

Shortest Path is :

v2 v3 v0



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 2

Enter destination vertex(-1 to quit): 0

Shortest Path is :

v2 v3 v0

Shortest distance is : 8

Enter source vertex : 3



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 3

Enter destination vertex(-1 to quit): 1

Shortest Path is :

v3 v1

Shortest distance is : 7

Enter source vertex : 1



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 1

Enter destination vertex(-1 to quit): 2

Shortest Path is :

v1 v2

Shortest distance is : 4

Enter source vertex : 1



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 1

Enter destination vertex(-1 to quit): 3

Shortest Path is :

v1 v2 v3

Shortest distance is : 5

Enter source vertex : 0



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 0

Enter destination vertex(-1 to quit): 3

Shortest Path is :

v0 v2 v3

Shortest distance is : 5

Enter source vertex : 3



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 3

Enter destination vertex(-1 to quit): 1

Shortest Path is :

v3 v1

Shortest distance is : 7

Enter source vertex : 2



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 2

Enter destination vertex(-1 to quit): 0

Shortest Path is :

v2 v3 v0

Shortest distance is : 8

Enter source vertex : 3



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter source vertex : 3

Enter destination vertex(-1 to quit): -1

Process exited after 113.9 seconds with return value 0
Press any key to continue . . .



Type here to search



C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe

Enter number of vertices : 4

Enter edge 1

enter origin vertex : 0

enter destination vertex : 1

Enter weight for this edge : 2

Enter edge 2

enter origin vertex : 0

enter destination vertex : 2

Enter weight for this edge : 4

Enter edge 3

enter origin vertex : 0

enter destination vertex : 3

Enter weight for this edge : 8



Type here to search



The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** Project3 - [Project3.dev] - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Build, Run, Stop, and others.
- Project Explorer:** Shows "Project3" under "Project".
- Code Editor:** Displays main.cpp with code for printing an adjacency matrix. The code uses nested loops to print the matrix and its transpose. A vertical line highlights the opening brace of the main function.
- Compiler Results:** Shows compilation results with 0 errors and 0 warnings. The output file is C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe, size 1.90373420715332 MiB, and compilation time 0.23s.
- Status Bar:** Shows Line: 17, Col: 21, Sel: 0, Lines: 174, Length: 4526, Insert, Done parsing in 0.031 seconds.
- Taskbar:** Shows the Windows Start button, a search bar with "Type here to search", and icons for File Explorer, Edge, File History, Task View, and others.
- System Tray:** Shows battery level, signal strength, volume, and system icons.

The screenshot shows the Dev-C++ IDE interface with the following details:

- Project Bar:** Project3 - [Project3.dev] - Dev-C++ 5.11
- File Menu:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Build, Run, Stop, and others.
- Compiler Selection:** TDM-GCC 4.9.2 64-bit Release
- Project Explorer:** Shows Project3 with a single file main.cpp selected.
- Code Editor:** Displays the main.cpp source code. The code prints an adjacency matrix and uses a while loop to run Dijkstra's algorithm between user-specified vertices.

```
main.cpp
26 cout<<"\n\n      The adjacency matrix is \n\n\n      ";
27 cout<<"      ";
28 for( j=0; j<=n-1; j++)
29 cout<<"      v<<j<<" ;
30
31
32 cout<<"\n\n";
33 for(i=0; i<=n-1; i++)
34 {
35     cout<<"      v<<i<<" ;
36         for(j=0; j<=n-1; j++)
37             cout<<"      "<<adjacency[i][j]<<" ";
38         cout<<"\n\n";
39 }
40
41 while(1)
42 {
43     cout<<"\n\n\n\n\n\n\n\n\n\n      Enter source vertex :      ";
44     cin>>s;
45
46 Dijkstra(s);
47     cout<<"\n\n\n\n\n\n      Enter destination vertex(-1 to quit):      ";
48     cin>>v;
49     cout<<"\n\n\n\n\n\n";
```

- Compiler Bar:** Compiler, Resources, Compile Log, Debug, Find Results, Close
- Compilation Results:** Shows compilation results with 0 errors and 0 warnings, outputting q3\Project3.exe.
- Status Bar:** Line: 17, Col: 21, Sel: 0, Lines: 174, Length: 4526, Insert, Done parsing in 0.031 seconds
- Taskbar:** Icons for File Explorer, Task View, Start, Taskbar settings, and a search bar.

The screenshot shows the Dev-C++ IDE interface. The top menu bar includes File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help, and a toolbar with various icons. The title bar says "Project3 - [Project3.dev] - Dev-C++ 5.11". The left sidebar shows the project structure with "Project3" selected under "Classes". The main editor window displays the following C++ code for a Dijkstra's algorithm implementation:

```
main.cpp
62     }
63 }
64
65 void Dijkstra( int s )
66 {
67     int i, current;
68
69     for(i=0; i<n; i++)
70     {
71         ancestors[i] = NIL;
72         path[i] = 9999;
73         status[i] = ZERO;
74     }
75
76     path[s] = 0;
77
78     while(1)
79     {
80         current = temp( );
81
82         if( current == NIL )
83             return;
84     }
85 }
```

The bottom panel shows the "Compiler" tab with the message "Compilation results...". It lists compilation statistics: 0 errors, 0 warnings, output filename as "C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe", output size of 1.90373420715332 MiB, and compilation time of 0.23s.

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project3

main.cpp

```
78
79     while(1)
80     {
81
82         current = temp();
83
84         if( current == NIL )
85             return;
86
87         status[current] = ONE;
88
89         for(i=0; i<n; i++)
90         {
91
92             if ( adjacency[current][i] !=0 && status[i] == ZERO )
93                 if( path[current] + adjacency[current][i] < path[i] )
94                 {
95                     ancestors[i] = current;
96                     path[i] = path[current] + adjacency[current][i];
97
98                 }
99
100            }
101
102        }
103
104    }
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe
- Output Size: 1.90373420715332 MiB
- Compilation Time: 0.23s

Type here to search

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TDM-GCC 4.9.2 64-bit Release

Project Classes Debug

Project3

main.cpp

```
102
103
104 int temp( )
105 {
106     int i;
107     int min = 9999;
108     int k = NIL;
109     for(i=0;i<n;i++)
110     {
111         if(status[i] == ZERO && path[i] < min)
112         {
113             min = path[i];
114             k = i;
115         }
116     }
117     return k;
118
119
120
121 void Path(int s, int v )
122 {
123     int i,u;
124     int path[100];
125     int shortdist = 0;
```

Compiler Resources Compile Log Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe
- Output Size: 1.90373420715332 MiB
- Compilation Time: 0.23s

Type here to search

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** Project3 - [Project3.dev] - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Build, Run, Stop, and others.
- Project Explorer:** Shows "Project3" under "Project".
- Code Editor:** Displays main.cpp with C++ code for a shortest path algorithm using Breadth-First Search (BFS). The code includes comments explaining the variables and logic. It uses global variables like `i`, `path`, `shortdist`, and `count`. It also uses local variables like `u` and `v` within functions.
- Status Bar:** Shows compilation results:
 - Compilation results...
 - Errors: 0
 - Warnings: 0
 - Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe
 - Output Size: 1.90373420715332 MiB
 - Compilation Time: 0.23s
- Bottom Bar:** Shows system icons for taskbar, search bar, and date/time (19:59, 22-05-2021).

The screenshot shows the Dev-C++ IDE interface with the following details:

- Title Bar:** Project3 - [Project3.dev] - Dev-C++ 5.11
- Menu Bar:** File, Edit, Search, View, Project, Execute, Tools, AStyle, Window, Help
- Toolbar:** Includes icons for New, Open, Save, Build, Run, Stop, and others.
- Project Explorer:** Shows "Project3" under "Project".
- Code Editor:** The main window displays the following C++ code:

```
main.cpp
146 void graph()
147 {
148     int i,max_edges,origin,destin, wt;
149
150     cout<<"\n\n\n      Enter number of vertices : ";
151     cin>>n;
152     max_edges = n*(n-1);
153
154     for(i=1;i<=max_edges;i++)
155     {
156         cout<<"\n\n\n      Enter edge <<i<< " ;
157         cout<<"\n\n      enter origin vertex : ";cin>>origin;cout<<"\n\n";
158
159         if( (origin == -1) && (destin == -1) )
160             break;
161
162         cout<<"\n\n      Enter weight for this edge : ";
163         cin>>wt;
164
165         if( (origin >= n || destin >= n || origin<0 || destin<0)
166         {
167             cout<<"\nInvalid edge!\n";
168             i--;
169         }
170     }
171 }
```

- Compiler Tab:** Shows "Compilation results..." with the following output:
- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe
- Output Size: 1.90373420715332 MiB
- Compilation Time: 0.23s
- Status Bar:** Line: 109 Col: 1 Sel: 0 Lines: 173 Length: 4434 Insert Done parsing in 0.016 seconds
- Taskbar:** Shows the Start button, task switcher, and pinned application icons for File Explorer, Edge, FileZilla, and others.

Project3 - [Project3.dev] - Dev-C++ 5.11

File Edit Search View Project Execute Tools AStyle Window Help

TIW-GCC 4.9.2 64-bit Release

Project Classes Debug

Project3

main.cpp

```
1 #include<iostream>
2 using namespace std;
3
4
5
6 #define ZERO 0
7 #define ONE 1
8 #define NIL -1
9
10 void Path(int s, int v );
11 void Dijkstra( int s);
12 int temp();
13 void graph();
14
15 int n;
16 int adjacency[100][100];
17 int ancestrons[100];
18 int path[100];
19 int status[100];
20
21 int main()
22 {
23     int s,v,i,j;
```

Compiler Resources CompileLog Debug Find Results Close

Compilation results...

- Errors: 0
- Warnings: 0
- Output Filename: C:\Users\Navin Ashok\Desktop\graphs\q3\Project3.exe
- Output Size: 1.90373420715332 MiB
- Compilation Time: 0.23s

Type here to search

Line: 17 Col: 21 Sel: 0 Lines: 174 Length: 4526 Insert Done parsing in 0.031 seconds

19:58 22-05-2021