shortestDistance

Your task is to write a function, shortestDistance, that returns the number of edges on the shortest path between two vertices in the given graph. If there is no path between the two vertices, return -1.

Download

Click here to download a zip of the files.

The Files

Graph.c	Contains the implementation of a graph ADT
Graph.h	Contains the interface of the graph ADT
Queue.c	Contains the implementation of a queue ADT
Queue.h	Contains the interface of the queue ADT
testShortestDistance.c	Contains the main function, which reads in a graph from standard input, calls shortestDistance for each pair of vertices read in, and prints out the results.
shortestDistance.c	Contains shortestDistance, the function you must implement
Makefile	A makefile to compile your code
tests/	A directory containing the inputs and expected outputs for some basic tests
autotest	A script that uses the tests in the tests directory to autotest your solution. You should only run this after you have tested your solution manually.

Examples

Your program should behave like these examples:

```
$ ./testShortestDistance
Enter number of vertices: 6
Enter number of edges: 5
Enter edges in the form v-w: 0-1 1-2 2-3 3-4 4-5

Graph: nV = 6
Edges:
 0: 0-1
 1: 1-0 1-2
 2: 2-1 2-3
 3: 3-2 3-4
 4: 4-3 4-5
 5: 5-4

Enter two vertices: 0 0
The shortest distance between vertices 0 and 0 is: 0
Enter two vertices: 0 3
The shortest distance between vertices 0 and 3 is: 3
Enter two vertices: 4 0
The shortest distance between vertices 4 and 0 is: 4
Enter two vertices: 1 5
The shortest distance between vertices 1 and 5 is: 4
Enter two vertices: (Ctrl + D)
```

```
$ ./testShortestDistance
Enter number of vertices: 10
Enter number of edges: 10
Enter edges in the form v-w: 0-1 0-2 1-3 1-6 2-9 3-4 3-5 5-7 5-9 7-8

Graph: nV = 10
Edges:
 0: 0-1 0-2
 1: 1-0 1-3 1-6
 2: 2-0 2-9
 3: 3-1 3-4 3-5
 4: 4-3
 5: 5-3 5-7 5-9
 6: 6-1
 7: 7-5 7-8
 8: 8-7
 9: 9-2 9-5

Enter two vertices: 0 7
The shortest distance between vertices 0 and 7 is: 4
Enter two vertices: 8 2
The shortest distance between vertices 8 and 2 is: 4
Enter two vertices: 5 6
The shortest distance between vertices 5 and 6 is: 3
Enter two vertices: (Ctrl + D)
```

```
$ ./testShortestDistance
Enter number of vertices: 10
Enter number of edges: 9
Enter edges in the form v-w: 0-1 1-2 1-3 2-4 2-5 3-5 3-6 7-8 8-9

Graph: nV = 10
Edges:
 0: 0-1
 1: 1-0 1-2 1-3
 2: 2-1 2-4 2-5
 3: 3-1 3-5 3-6
 4: 4-2
 5: 5-2 5-3
 6: 6-3
 7: 7-8
 8: 8-7 8-9
 9: 9-8

Enter two vertices: 6 4
The shortest distance between vertices 6 and 4 is: 4
Enter two vertices: 0 5
The shortest distance between vertices 0 and 5 is: 3
Enter two vertices: 0 8
There is no path between vertices 0 and 8
Enter two vertices: (Ctrl + D)
```

Testing

You can test your program manually by compiling your code using `make`, and then running `./testShortestDistance`, as shown above. After you are satisfied with your solution, you can autotest it by running `./autotest`. This will run some basic tests on your program.