numWithin

Your task is to write a function, numWithin, that determines the number of vertices that are within a given distance of the given source vertex.

Download

Click here to download a zip of the files.

The Files

Graph.c	Contains the implementation of basic graph functions
Graph.h	Contains the definition of the Graph data structure and function prototypes
Queue.c	Contains the implementation of a queue ADT
Queue.h	Contains the interface of the queue ADT
testNumWithin.c	Contains the main function, which reads in a graph from standard input, calls numWithin for each source vertex and distance read in, and prints out the results.
numWithin.c	Contains numWithin, 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:

```
$ ./testNumWithin
Enter number of vertices: 11
Enter number of edges: 10
Enter edges in the form v-w: 0-1 1-2 2-3 3-4 0-5 5-6 6-7 0-8 8-9 0-10

Graph: nV = 11
Edges:
 0: 0-1 0-5 0-8 0-10
 1: 1-0 1-2
 2: 2-1 2-3
 3: 3-2 3-4
 4: 4-3
 5: 5-0 5-6
 6: 6-5 6-7
 7: 7-6
 8: 8-0 8-9
 9: 9-8
10: 10-0

Enter the source vertex and maximum distance: 0 5
numWithin(g, 0, 5) = 11
Enter the source vertex and maximum distance: 0 4
numWithin(g, 0, 4) = 11
Enter the source vertex and maximum distance: 0 3
numWithin(g, 0, 3) = 10
Enter the source vertex and maximum distance: 0 2
numWithin(g, 0, 2) = 8
Enter the source vertex and maximum distance: 0 1
numWithin(g, 0, 1) = 5
Enter the source vertex and maximum distance: 0 0
numWithin(g, 0, 0) = 1
Enter the source vertex and maximum distance: (Ctrl + D)
```

```
$ ./testNumWithin
Enter number of vertices: 11
Enter number of edges: 12
Enter edges in the form v-w: 0-1 0-2 0-6 0-7 1-2 2-3 2-5 3-4 6-7 7-8 7-10 8-9

Graph: nV = 11
Edges:
 0: 0-1 0-2 0-6 0-7
 1: 1-0 1-2
 2: 2-0 2-1 2-3 2-5
 3: 3-2 3-4
 4: 4-3
 5: 5-2
 6: 6-0 6-7
 7: 7-0 7-6 7-8 7-10
 8: 8-7 8-9
 9: 9-8
10: 10-7

Enter the source vertex and maximum distance: 1 1
numWithin(g, 1, 1) = 3
Enter the source vertex and maximum distance: 1 2
numWithin(g, 1, 2) = 7
Enter the source vertex and maximum distance: 1 3
numWithin(g, 1, 3) = 10
Enter the source vertex and maximum distance: 1 4
numWithin(g, 1, 4) = 11
Enter the source vertex and maximum distance: 4 1
numWithin(g, 4, 1) = 2
Enter the source vertex and maximum distance: 4 4
numWithin(g, 4, 4) = 8
Enter the source vertex and maximum distance: (Ctrl + D)
```

Testing

You can test your program manually by compiling your code using `make`, and then running `./testNumWithin`, 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.