

# breadthFirstSearch

Your task is to write a function, `breadthFirstSearch`, that performs a breadth first search on a graph starting at the given vertex. It should print out the vertices as they are visited. If a vertex has multiple neighbours, visit the neighbour with the smallest vertex number first.

## Download

Click [here](#) to download a zip of the files.

## The Files

<b>Graph.c</b>	Contains the implementation of a graph ADT
<b>Graph.h</b>	Contains the interface of the graph ADT
<b>Queue.c</b>	Contains the implementation of a queue ADT
<b>Queue.h</b>	Contains the interface of the queue ADT
<b>testBreadthFirstSearch.c</b>	Contains the main function, which reads in a graph and a starting vertex from standard input, and then calls <code>breadthFirstSearch</code> .
<b>breadthFirstSearch.c</b>	Contains <code>breadthFirstSearch</code> , the function you must implement
<b>Makefile</b>	A makefile to compile your code
<b>tests/</b>	A directory containing the inputs and expected outputs for some basic tests
<b>autotest</b>	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:

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

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

Enter src: 0
Breadth first search starting at vertex 0: 0 1 7 2 5 3 8 4 6
```

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

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

Enter src: 3
Breadth first search starting at vertex 3: 3 2 4 1 5 0 6 7
```

```
$ ./testBreadthFirstSearch
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 src: 6
Breadth first search starting at vertex 6: 6 3 1 5 0 2 4
```

## Testing

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