In all the sample runs below the the user input is just the file name.

----------Sample run showing execution when file not found.

```
gcc -g -std=c99 courses_graph.c
valgrind --leak-check=full ./a.out
==25615== Memcheck, a memory error detector
==25615== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==25615== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
==25615== Command: ./a.out
==25615==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
Enter filename: badName
Could not open file badName. Exit

Failed to read from file. Program will terminate.

==25615==
==25615== HEAP SUMMARY:
==25615==     in use at exit: 0 bytes in 0 blocks
==25615==   total heap usage: 1 allocs, 1 frees, 568 bytes allocated
==25615==
==25615== All heap blocks were freed -- no leaks are possible
==25615==
==25615== For counts of detected and suppressed errors, rerun with: -v
==25615== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

----------Sample run when there is a cycle in the graph.

```
valgrind --leak-check=full ./a.out
==27653== Memcheck, a memory error detector
==27653== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==27653== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
==27653== Command: ./a.out
==27653==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
Enter filename: cycle0.txt
Number of vertices in built graph:  N = 3
Vertex - course name correspondence
0 - c100
1 - c300
2 - c200

Adjacency matrix:
    |   0   1   2
-----------------
   0|   0   1   1
   1|   0   0   0
   2|   1   1   0
```

```
There was at least one cycle. There is no possible ordering of the courses.

==27653==
==27653== HEAP SUMMARY:
==27653==     in use at exit: 0 bytes in 0 blocks
==27653==   total heap usage: 9 allocs, 9 frees, 745 bytes allocated
==27653==
==27653== All heap blocks were freed -- no leaks are possible
==27653==
==27653== For counts of detected and suppressed errors, rerun with: -v
==27653== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

----------Sample run when there is no cycle in the graph.

```
valgrind --leak-check=full ./a.out
==27818== Memcheck, a memory error detector
==27818== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==27818== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
==27818== Command: ./a.out
==27818==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
Enter filename: data0.txt
Number of vertices in built graph:  N = 3
Vertex - course name correspondence
0 - c100
1 - c300
2 - c200

Adjacency matrix:
    |   0   1   2
-----------------
   0|   0   1   1
   1|   0   0   0
   2|   0   1   0


Order in which to take courses:
1. - c100 (corresponds to graph vertex 0)
2. - c200 (corresponds to graph vertex 2)
3. - c300 (corresponds to graph vertex 1)

==27818==
==27818== HEAP SUMMARY:
==27818==     in use at exit: 0 bytes in 0 blocks
==27818==   total heap usage: 9 allocs, 9 frees, 745 bytes allocated
==27818==
==27818== All heap blocks were freed -- no leaks are possible
==27818==
==27818== For counts of detected and suppressed errors, rerun with: -v
==27818== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

----------Sample run for same courses as in data0.txt, but given in a different order in the file (data0_rev.txt).

```
valgrind --leak-check=full ./a.out
==5170== Memcheck, a memory error detector
==5170== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==5170== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
==5170== Command: ./a.out
==5170==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
Enter filename: data0_rev.txt
Number of vertices in built graph:  N = 3
Vertex - course name correspondence
0 - c300
1 - c200
2 - c100

Adjacency matrix:
    |   0   1   2
----------------
   0|   0   0   0
   1|   1   0   0
   2|   1   1   0


Order in which to take courses:
1. - c100 (corresponds to graph vertex 2)
2. - c200 (corresponds to graph vertex 1)
3. - c300 (corresponds to graph vertex 0)

==5170==
==5170== HEAP SUMMARY:
==5170==     in use at exit: 0 bytes in 0 blocks
==5170==   total heap usage: 9 allocs, 9 frees, 745 bytes allocated
==5170==
==5170== All heap blocks were freed -- no leaks are possible
==5170==
==5170== For counts of detected and suppressed errors, rerun with: -v
==5170== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```

----------Sample run for data1.txt

```
valgrind --leak-check=full ./a.out
==479== Memcheck, a memory error detector
==479== Copyright (C) 2002-2017, and GNU GPL'd, by Julian Seward et al.
==479== Using Valgrind-3.15.0 and LibVEX; rerun with -h for copyright info
==479== Command: ./a.out
==479==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
```

```
Enter filename: data1.txt
Number of vertices in built graph:  N = 8
Vertex - course name correspondence
0 - cs40
1 - cs20
2 - cs800
3 - cs50
4 - MATH101
5 - MATH1421
6 - cs10
7 - cs30

Adjacency matrix:
    |   0   1   2   3   4   5   6   7
-----------------------------------
   0|   0   0   0   1   0   0   0   0
   1|   0   0   0   0   0   0   0   0
   2|   0   0   0   0   0   0   0   0
   3|   0   0   0   0   0   0   0   0
   4|   0   0   0   0   0   1   0   0
   5|   1   0   0   0   0   0   0   0
   6|   1   0   0   0   0   0   0   1
   7|   1   0   0   0   0   0   0   0

Order in which to take courses:
1. - cs10 (corresponds to graph vertex 6)
2. - cs30 (corresponds to graph vertex 7)
3. - MATH101 (corresponds to graph vertex 4)
4. - MATH1421 (corresponds to graph vertex 5)
5. - cs800 (corresponds to graph vertex 2)
6. - cs20 (corresponds to graph vertex 1)
7. - cs40 (corresponds to graph vertex 0)
8. - cs50 (corresponds to graph vertex 3)

==479==
==479== HEAP SUMMARY:
==479==     in use at exit: 0 bytes in 0 blocks
==479==   total heap usage: 22 allocs, 22 frees, 7,248 bytes allocated
==479==
==479== All heap blocks were freed -- no leaks are possible
==479==
==479== For lists of detected and suppressed errors, rerun with: -s
==479== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 0 from 0)
```

----------Sample run for the graph on the slide for topological sorting.

Note that the program produced the same start and finish time and the same vertex order (6,4,2,3,0,1,5). For grading and correctness checking, make sure you have the same mapping between course names and vertex numbers as shown in this sample run.

```
valgrind --leak-check=full ./a.out
==26380== Memcheck, a memory error detector
==26380== Copyright (C) 2002-2009, and GNU GPL'd, by Julian Seward et al.
==26380== Using Valgrind-3.5.0 and LibVEX; rerun with -h for copyright info
==26380== Command: ./a.out
==26380==

This program will read, from a file, a list of courses and their
prerequisites and will print the list in which to take courses.
Enter filename: slides.txt
Number of vertices in built graph:  N = 7
Vertex - course name correspondence
0 - CSE1000
1 - CSE1001
2 - CSE1002
3 - CSE1003
4 - CSE1004
5 - CSE1005
6 - CSE1006

Adjacency matrix:
    |   0   1   2   3   4   5   6
--------------------------------
   0|   0   1   0   0   0   1   0
   1|   0   0   0   0   0   1   0
   2|   0   0   0   1   0   1   0
   3|   0   0   0   0   0   0   0
   4|   0   1   0   0   0   1   0
   5|   0   0   0   0   0   0   0
   6|   0   0   1   1   0   0   0


Order in which to take courses:
1. - CSE1006 (corresponds to graph vertex 6)
2. - CSE1004 (corresponds to graph vertex 4)
3. - CSE1002 (corresponds to graph vertex 2)
4. - CSE1003 (corresponds to graph vertex 3)
5. - CSE1000 (corresponds to graph vertex 0)
6. - CSE1001 (corresponds to graph vertex 1)
7. - CSE1005 (corresponds to graph vertex 5)

==26380==
==26380== HEAP SUMMARY:
==26380==     in use at exit: 0 bytes in 0 blocks
==26380==   total heap usage: 17 allocs, 17 frees, 1,093 bytes allocated
==26380==
==26380== All heap blocks were freed -- no leaks are possible
==26380==
==26380== For counts of detected and suppressed errors, rerun with: -v
==26380== ERROR SUMMARY: 0 errors from 0 contexts (suppressed: 4 from 4)
```