**CSCE 221 Cover Page   
Program Assignment #5**

Ryan Wenham 627002098

Ryanjw0903@tamu.edu

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero, read more in the Aggie Honor System Office <http://aggiehonor.tamu.edu/>

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type of sources |  |  |  |  |  |
|  |  |  |  |  |  |
| People | Sam, Marrisa |  |  |  |  |
|  |  |  |  |  |  |
| Web pages (provide URL) | GeeksofGeeks.com |  |  |  |  |
|  |  |  |  |  |  |
| Printed material |  |  |  |  |  |
|  |  |  |  |  |  |
| Other Sources |  |  |  |  |  |
|  |  |  |  |  |  |

I certify that I have listed all the sources that I used to develop the solutions/code to the submitted work.

“*On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work.*”

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Your Name (signature) | Ryan Wenham |  | Date | 4/27/2020 |

Program Assignment 5 was to teach us how to understand the usefulness of graphs in coding and to be able to make one with an adjacency matrix. Then to use that matrix to find the topographical sort of that matrix. To do this sort I used the Kahn’s algorithm, with a big-O notation of O(vertices+edges). The algorithm works by first computing the n-degree of each vertex. Next add all n-degrees of zero to a queue. Remove vertex from queue by pop and decrease the n-degree of each of it connect vertices. If any become equal to zero add to the queue. Repeat this process till the queue is empty. You can also use a stack and treat it like the queue with a DFS method to find the topographical sort. To check for a cycle, I count each vertex that is visited. If the total count does not equal the length of the vertex list then I now not all vertex was reached so there had to be a cycle.

Features I used where the vectors to store the adjacency matrix and list of vertexes. The built-in list feature was used to make the adjacency matrix so for each vertex to point to other vertexes. To read in the values I use get line and string stream to turn strings into ints as well.

Below are the test cases 1-4:

A close up of a logo

Description automatically generated

A close up of a logo

Description automatically generated

A picture containing bird, flower

Description automatically generated

A close up of a logo

Description automatically generated