CSCE 221 Cover Page
Programming Assignment #5

First Name: Cora Last Name: English UIN: 127007634

User Name: coraenglish E-mail Address: coraenglish@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. According to the University Regulations, Section 42, scholastic dishonesty are including: acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)	http://www.cplusplus.com/reference/queue/queue/	http://www.cplusplus.com/reference/vector/vector/	http://www.cplusplus.com/reference/list/list/
Printed material			
Other Sources			

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

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

Electronic signature Cora English Date 4/26/2020

Part 3 (20 points) – due April 27

- (15 points) description of your implementation, C++ features used, assumptions on input data.
  - Why does the algorithm use a queue? Can we use a stack instead?
    - This algorithm uses a queue in order to correctly order the elements that have indegrees of 0. A queue is used instead of a stack because of its FIFO property. The order one enqueues items is the correct order in which to traverse the graph. If a stack was used instead, popping off of the end would interfere with the ordering. Granted, this is still a correct ordering, it just won't be guaranteed to match the ordering the queue provides.
  - Can you explain why the algorithm detects cycles?
    - The algorithm detects cycles by keeping track of the number of enqueued elements and comparing that to the total amount of vertices. When a cycle is present the counter will be less than the number of vertices. In the example data provided, 1, 2, and 3 are all eventually enqueued and then dequeued, but no other vertices are. This is beacuse of the cycle present. Not every vertex will be reached because once 3 is popped from the queue, there is no other vertex with an indegree of 0 to be placed in the queue, so the loop stops and the if statement is met which exits the program.
  - What is the running time for each function? Use the Big-O notation asymptotic notation and justify your answer.
    - **buildGraph** has a runtime of O($V+E$). The function first starts by reading in all of the data from the file into nodes_list. Then, a nested while loop is encountered. The outer loop traverses through all of the integers in nodes_list (V) and then the inner loop adds all of the edges in the adjacency list into the associated linked list (E).
    - **displayGraph** has a run time of O($V+E$). This function has a nested for loop. This first will print out the vertices starting from 1 while the second will print out all of the edges from that vertex in the form of an adjacency list.
    - **topological_sort** has a run time of O($V+E$). The while loop will continue running until the queue is empty, meaning all vertices will eventually be enqueued or there is a cycle in the graph. The for loop inside the while loop goes through the adjacency list at the current vertex that has been dequeued and decrements the indegree of the corresponding vertices by 1.
    - **compute_indegree** has a run time of O($V+E$). The nested for loop goes through each vertex in the adjacency list, then increments the appropriate indegree in the vertices vector by 1.
    - **print_top_sort** has a run time of O(V^2). This is because the function has two for loops: the first loop starts at 1 and goes through all of the vertices (V times) and the second starts at the first vertex in the vector that holds indegree and top_num info until the top_num == i (a potential V times).

- (5 points) test your program for correctness using the four cases below:

  Case 1: Use the example (`input.data`) provided in the description of the problem.

  Answer: 1 2 3 4 7 6 5

  Case 2: Samantha plans her course schedule. She is interested in the following eight courses: CSCE121, CSCE222, CSCE221, CSCE312, CSCE314, CSCE313, CSCE315, and CSCE411. The course prerequisites are:

course	#	prerequisites	
CSCE121:	1	(none)	
CSCE222:	2	(none)	
CSCE221:	3	CSCE121	CSCE222
CSCE312:	4	CSCE221	
CSCE314:	5	CSCE221	
CSCE313:	6	CSCE221	
CSCE315:	7	CSCE312	CSCE314
CSCE411:	8	CSCE222	CSCE221

  Find a sequence of courses that allows Samantha to satisfy all the prerequisites. Assume that she can only take one class at a time. The input file for this case is provided (`input2.data`)

  Answer: 1 2 3 4 5 6 8 7 or 121 222 221 312 314 313 411 315

  Case 3: Samantha loves foreign languages and wants to plan her course schedule. She is interested in the following nine courses: LA15, LA16, LA22, LA31, LA32, LA126, LA127, LA141, and LA169. The course prerequisite are:

course	#	prerequisites	
LA15:	1	(none)	
LA16:	2	LA15	
LA22:	3	(none)	
LA31:	4	LA15	
LA32:	5	LA16	LA31
LA126:	6	LA22	LA32
LA127:	7	LA16	
LA141:	8	LA22	LA16
LA169:	9	LA32	

  Find a sequence of courses that allows Samantha to satisfy all the prerequisites. Assume that she can only take one class at a time.

  Answer: 1 3 2 4 7 8 5 6 9 or LA15 LA22 LA16 LA31 LA127 LA141 LA32 LA126 LA169

  Case 4. Create a directed graph with cycles and test your program. There is one such a file provided (`input-cycle.data`).

  Answer: Cycle found, so no order was output.