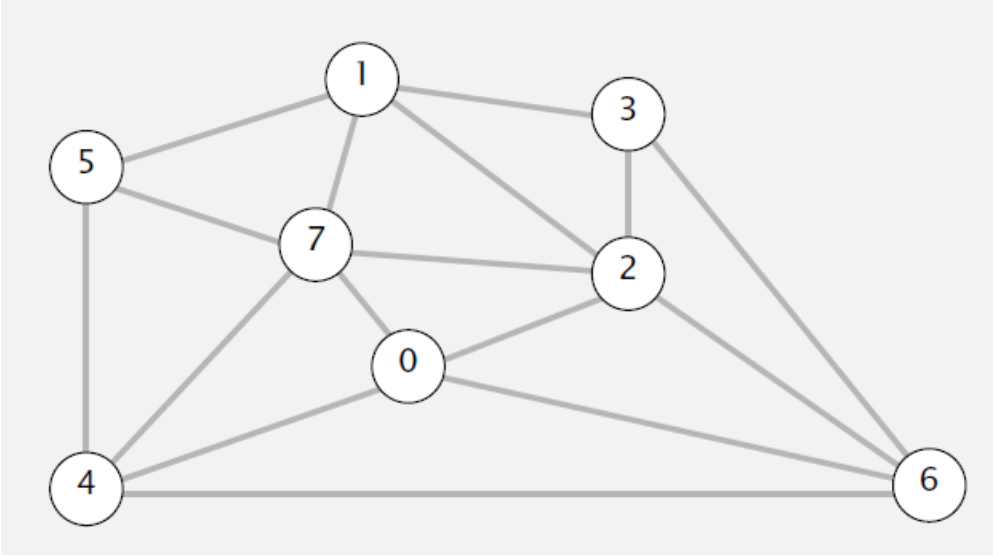
CS 101 **Project 5** **Fall 2016** **Due 11/27**

For this project you will be given an undirected graph, G, where every edge is colored either Crimson or White. You will also be given an additional parameter, x. Your job is to either find a spanning tree of G that contains exactly x Crimson edges or report that no spanning tree of G contains x Crimson edges.

The input file name will be given as the first command line parameter and the parameter x will be the second command line parameter. The format of the input file will be as follows:
The first line will be the number of vertices followed by the number of edges. Each of the following lines represents an edge, given as the two integer vertex numbers followed by either a capitol C or capitol W for Crimson or White.

```
8  16
0 7 C
2 3 W
1 7 C
0 2 C
5 7 W
1 3 W
1 5 C
2 7 C
4 5 W
1 2 W
4 7 W
0 4 C
6 2 W
3 6 W
6 0 C
6 4 C
```



Hint(s):

You should modify Prim's Algorithm to use a priority queue that you build. This priority queue should support extracting Crimson first and extracting White first. The extract and insert operations should run in O(1) time. This make Prim's algorithm run in O(V + E) time and your overall program should also run in O(V + E) time.

First run Prim's algorithm with White as the priority. This will determine the minimum number of Crimson edges needed in any spanning tree of G. Use those crimson edges as a starting point and run Prim's algorithm again, but with Crimson edges as the priority. Once you reach x Crimson edges, you may complete the spanning tree with White edges.

The output should be a spanning tree of G given as a list of edges, one per line, or the statement "No spanning tree with $x Crimson edges exists" where $x is replaced by the value of the parameter x. Make sure that this statement is exactly as it appears. No other output should appear.

In the example above, there are spanning trees of the input graph containing 2,3,4,5, or 6 Crimson edges.

Please carefully read and follow all the following requirements:

- You must do your own work; you must not share any code with any other person. If you violate this rule, you may receive an invitation to the dean's office to discuss the penalties for academic misconduct.
- Make sure your program runs properly on the cs-intro.ua.edu server. Your program will be graded on that system.
- Submit your project in a zipfile that contains a Makefile and all your .cpp files and .h files. There should be no subdirectories or extra files. Typing "make" should build a project executable called "p5.exe".
- If you violate the requirements such that it breaks our grading script, your project will be assessed a significant point deduction, and extreme or multiple violations may cause the project to be considered ungradable.
- Submit your project on Blackboard by 11:59pm on the due date. Projects will be accepted up to three days late, but will be assessed a late penalty of 10% for each day late.
- Double-check your submitted project when you submit it. Errors discovered later cannot be fixed and resubmitted after the project is graded. Projects will not be re-graded unless an error is found in the grading script or in the input/output files used during grading.
- In particular, double-check that you've included all the latest versions of your files, and that all files are named correctly. Also double-check that all input/output works exactly as specified.
- It is your responsibility to make sure that the program you submit will run properly on the cs-intro.ua.edu server. We cannot debug and fix errors while grading projects for all the students in this course.