Programming Assignment III (due Friday, December 7, 2012)

In this project you will implement the procedures for solving the **Independent Set** problem, **K Coloring** problem and the **Graph Isomorphism** problem, following the discussions we had in class.

(a) Write a MAPLE procedure that takes as input a graph G (of arbitrary size) and a positive integer k, and outputs *true* if G has an independent set of size k, and $false$ otherwise.

(b) Write a MAPLE procedure that takes as input a graph G (of arbitrary size) and a positive integer k, and outputs *true* if G has a proper k-coloring, and $false$ otherwise.

(c) Write a MAPLE procedure that takes as input two graphs G_1 and G_2 outputs *true* if G_1 is isomorphic to G_2, and $false$ otherwise. Make sure you make the trivial check of whether they have the same number of vertices before you do more sophisticated calculations.

A few comments/suggestions :

1. Perhaps the trickiest bit of the assignment might be the fact that the number of variables (for your polynomial equations) depends on the input. There are a couple of ways to tackle this issue in MAPLE. For a hint, see how we set up the permutation matrix in class for the graph isomorphism.
2. You might want to use the $IsProper()$ command in the package *with*(*Groebner*) which returns *true* if a system of polynomial equations has at least 1 solution, and $false$ if it has no solutions. You are welcome to use any other MAPLE command of your choice.
3. Remember to use *with*(*GraphTheory*) and *with*(*LinearAlgebra*) packages.

(d) Use your codes to find the maximum size independent set in the *Petersen graph*. This special graph is included in the package *with*(*SpecialGraphs*) (you can see a picture of it by using *DrawGraph*()). Also show that the Petersen graph is 3-colorable but not 2-colorable. Also show the *Dodecahedron graph* (also available in *with*(*SpecialGraphs*)) is 3-colorable, but not 2-colorable.

(e) An **automorphism** of a graph $G = (V, E)$ is a map $\sigma : V \to V$ such that $(\sigma(i), \sigma(j)) \in E$ if and only if $(i, j) \in E$. Of course, every graph has at least one automorphism: the identity map $\sigma(i) = i$ where each vertex is mapped to itself. Some graphs have more interesting automorphisms. For example, convince yourself that the complete graph on n vertices (which means the graph which includes all possible $\binom{n}{2}$ edges) has $n!$ automorphisms.

Write a MAPLE procedure which takes as input any graph G (of arbitrary size) and **counts the number** of autmorphisms of this graph. You can call the procedure you wrote from part (c) above. Test your code on *PathGraph*() of various sizes, *CompleteGraph*() of various sizes, *CycleGraph*() of various sizes, and then some random graphs of various sizes. For the path, cycle and complete graphs, you might want to check that you are getting the right answer by checking it with a combinatorial argument (you dont have to include this argument in what you hand in).

Cautionary Note: Note that for the graph isomorphism (or automorphism) problems, if you have a graph on n vertices, then the problem sets up polynomial equations in n^2 variables. This means, with $n = 10$, you are feeding a problem with 100 variables to MAPLE. This might smoke MAPLE, so restrict yourself to smaller sized graphs. For large graphs, you should give MAPLE 15-20 minutes of computing time before killing it. Play around with what sizes give you the answers in a reasonable time.