## Part 1. Description of how to run my code and some example outputs.

Note: my used version of Prolog is SWI-Prolog

1.1. Substitute the input part with any directed graph you want to input with the format 'arc(vertex1, vertex2, weight)' in the file 'myPath.pl'

1.2. Open 'myPath.pl' and consult it in SWI-Prolog-Editor

1.3. Input the query 'path(source, destination, P)' end click ENTER, and then you can get the result. The result is in the format 'P = [Minimum_Cost, [Shortest Path in Vertices]]'

***Example 1.***

Take the toy graph in the assignment PDF as the 1st example:

```
arc(m,p,8).
arc(q,p,11).
arc(q,m,5).
arc(k,q,3).
```

Input the query 'path(k, p, P)' and the result will be outputted like this:

```
17 ?- path(k,p,P).
P = [14, [k, q, p]]
```

***Example 2.***

Try a more complicated graph such as this one:

```
arc(a,b,10).
arc(a,e,2).
arc(a,d,10).
arc(e,b,1).
arc(d,e,1).
arc(b,c,3).
arc(b,f,6).
arc(e,f,7).
arc(d,f,6).
arc(f,c,2).
arc(c,g,4).
arc(f,g,1).
```

Input some queries and the corresponding are as below:

```
3 ?- consult('myPath.').
% myPath. compiled 0.00 sec, 17 clauses
true.

4 ?- path(a,g,P).
P = [10, [a, e, b, c, g]] ;
false.

5 ?- path(a,f,P).
P = [9, [a, e, b, f]] ;
false.

6 ?- path(a,c,P).
P = [6, [a, e, b, c]] ;
false.

7 ?-
```

## Part 2. Reference


1. SWI Prolog documentation: http://www.swi-prolog.org/pldoc/index.html

2. A nice Prolog tutorial: https://www.csupomona.edu/~jrfisher/www/prolog_tutorial/contents.html#2

3. A blog introducing how to use 'List' in Prolog: http://www.cnblogs.com/jv9/articles/2199232.html


## Part 3. A statement of the Aggie Code of Honor.

An Aggie does not lie, cheat or steal or tolerate those who do.