1. Get a copy of the sample program `ngraph.c`. Compile and link the program by typing:
   \texttt{make ngraph < cr >}
   
   This program analyzes a network graph which is entered via a datafile. A sample datafile
   is `ngraph.dat` which is the example graph we looked at in class with eight nodes.

2. I wrote the program some time ago and, as user interfaces go, it has some rough edges.
   Run and it will prompt you for the name of the datafile. It then gives you seven
   options. Use \texttt{m} to print the matrix of metric (distance) values for the edges of the graph.
   Use \texttt{s} to run Dijkstra’s algorithm for shortest path, entering the two nodes \textit{by number}. It
   will show you how the “probing” goes and which nodes are being made “permanent” in
   bold. Use \texttt{S} to print out a full table of shortest paths.

   The datafile accepts capacity numbers as well as metric distances and you can find things
   like \textit{maximum flow} (which is the same as \textit{minimum cut}).

\textbf{Assignment} Make your own network graph with at least 8 nodes and run Dijkstra’s
algorithm by hand to find shortest paths from one fixed node. Make a datafile for your
graph and run the algorithm with `ngraph` to check your work. Email me your (text)
datafile and your table of shortest paths.