Brandeis Map Project

Implement a map utility for the Brandeis campus. Your program will ask for a begin location, a finish location, whether using a skateboard, and whether to minimize distance or time (constants are provided to define different travel times for walk flat, walk up hill, skateboard down hill, etc.) Although you may discuss program design with others, all code you submit must be written by you (no additional #includes), with the exception of the following files that are provided for this assignment:

BrandeisMapLabeled.jpg, **BrandeisMapLabeledCropped.jpg**.

MapDataVertices.txt: A list of vertices, one line for the information associated with each vertex.
MapDataEdges.txt: A list of directed edges, one line for the information associated with each edge.

Map.c: C program that is missing these portions (you may also have to change the string "PATH"):
Graph adjacency list data structure.
Heap data structure.
Dijkstra's single source shortest path algorithm ($O(m\log(n)$ time and $O(n+m)$ space).

Files included by Map.c:
MapTime.h: Function *Time(i)* to be called by your code.
MapInputData.h: Map data input functions called from main (no need to look at these).
MapInputUser.h: User input functions called from main (no need to look at these).
MapOutput.h: The function *PrintLeg* to be called by your code.

Display.m: A MatLab program that will display a route for you on the Brandeis map.

Sample Outputs: Some .txt and .jpg files showing sample output from a working program.

Solutions.txt: Output from the tests (below), using double precision to compute times.

What to pass in (email a zip archive):

Map.c - Should be well commented and organized; ***begin this file with your name.*** Although in practice, one might want to put code in multiple files (and perhaps use a different programming language), to simplify grading, you must make a C program by editing Map.c. Before you pass it in, you must check that it compiles without any errors or warnings under *standard ANSI C, on a machine in the COSCI Lounge*, and produce your output file on that machine. For example, from a terminal window, you can create the program "Map" by doing

```
gcc -ansi -Wall -o Map Map.c
```

and then run your program by doing:

```
./Map
```

ReadMe.txt - A plain text file that describes how your code works; ***begin this file with your name***.

Output.txt - Your output for these test inputs:

1. U14 L24 board=y time=y
2. U14 L24 board=y time=n
3. U14 L24 board=n time=n
4. U37 L5 board=y time=y
5. U37 L5 board=y time=n
6. U37 L5 board=n time=n
7. U40 + board=y time=y
8. U40 + board=n time=n
9. U17 L36 board=y time=y
10. L1 A4 board=n time=y
11. L36 $ board=n time=y
12. ! = board=n time=n