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. 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 (also see the ReadMe file):

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:
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).

Tour.h: Included by Map.c; reserved for Part 2 of this assignment.

Files included by Map.c (NOT TO BE MODIFIED)
Map.h: Map data and parameters.
MapData.h: Functions to input the map data and compute time (no need to look at these).
MapInput.h: User input functions called from main (no need to look at these).
MapOutput.h: User output functions (no need to look at these).
MapPATH.h: Directory path; the comment there explains how you may have to change it.

Display.m, DisplayCropped.m: MatLab programs to display a route on the Brandeis map.

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

Solutions.txt: Output from the test inputs listed below.

What to pass in (e.g., if you are *John Smith*, pass in the folder *Smith_John* with these files):

ReadMe.txt: A plain text file that describes how your code works and / or problems you had.

Map.c: Although in practice, one might want to use multiple files (and perhaps a different programming language), to simplify grading, you must make a single C program file by editing Map.c, and it must be that it compiles without any errors or warnings using *standard ANSI C*.

Output.txt: In a terminal window in the COSCI lounge, create the program *Map* by doing:

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

Then do ./Map and save the outputs from these tests in *Output.txt*:

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