



JAVAD IBRAHIMLI

040210932

---

# **EHB 208E HOMEWORK REPORT**

---

**2023 SPRING**

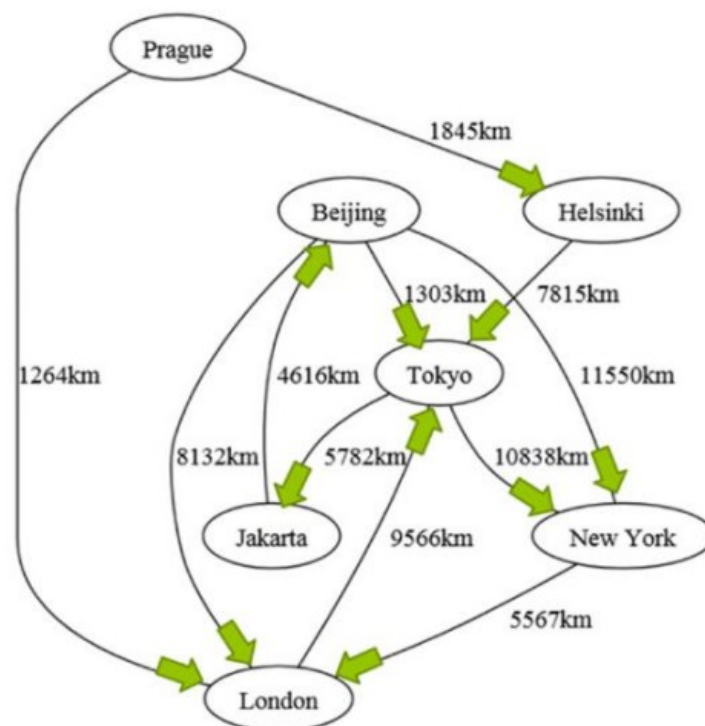
# Table of Contents

1. Introduction.....	3
1.1 Brief explanation of the code and its purpose.....	3
1.2 Overview of the programming language used and its version.....	4
2. How To Run This Code.....	5
2.1 Set your input file and directory correctly.....	5
2.2 Run the code.....	5
2.3 Terminal Output.....	6

## 1. Introduction

### 1.1 Brief explanation of the code and its purpose

Write a “C” program using the graph below to find and print the path between the two cities using the depth-first search algorithm in a specific order (always choose the city with the shortest distance). The graph will be automatically imported from the given text file. Thus, it is possible to define any graph in your code with the given text file. Your code should generate the desired output file.



**Important: Your code will be compiled with gcc or g++ and your grade will be automatically given. So, the text in the output file should be correctly typed (Not even extra spaces allowed. It is type-strict as the language:).**

**Your code should be properly commented. Uncommented code will get partial credit.**

**Your code should input the graph and queries from the given text file. Otherwise, your code will get partial credit.**

**You need to do your assignment alone. Code sharing among students or using code from any other source is not allowed.**

*For this graph, the given input text file contains:*

7 11

Prague Helsinki 1845

Prague London 122264

Beijing London 8132

Beijing Tokyo 1303

Beijing NewYork 11550

Helsinki Tokyo 7815

Tokyo Jakarta 5782

Tokyo NewYork 10838

Jakarta Beijing 4616

NewYork London 5567

London Tokyo 9566

Prague London

London London

London Prague

where in the first line, the number of nodes and the number of edges are given. In the following 11 lines, edges are given in the format of {SOURCE\_NODE, DESTINATION\_NODE, LENGTH}. The last three lines represent the questioned paths.

The distance is the length of this path. Your code should run on the terminal when typing

**`./hw2 -i input.txt -o output.txt`**

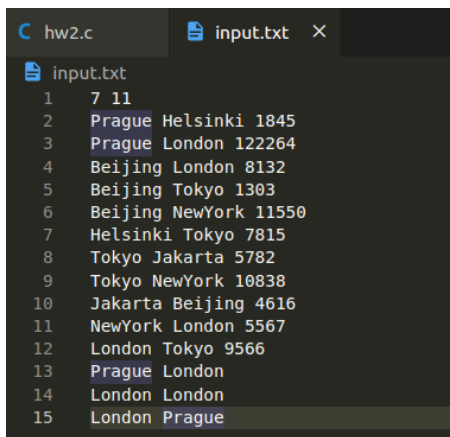
and should generate the desired output file. Here input.txt and output.txt are optional parameters that can be replaced with any input/output file path.

## **1.2 Overview of the programming language used and its version**

C programming language is used to write this code. Standart C++ headers is used in this code.

## 2. How To Run This Code

### 2.1 Set your input file and directory correctly



```
1 7 11
2 Prague Helsinki 1845
3 Prague London 122264
4 Beijing London 8132
5 Beijing Tokyo 1303
6 Beijing NewYork 11550
7 Helsinki Tokyo 7815
8 Tokyo Jakarta 5782
9 Tokyo NewYork 10838
10 Jakarta Beijing 4616
11 NewYork London 5567
12 London Tokyo 9566
13 Prague London
14 London London
15 London Prague
```

This is the recommended test input.txt file. I will use this input file in my code.

Goal of lines in this example:

1- It shows how many cities and roads are there in the map.

2-12 – Distance between cities.

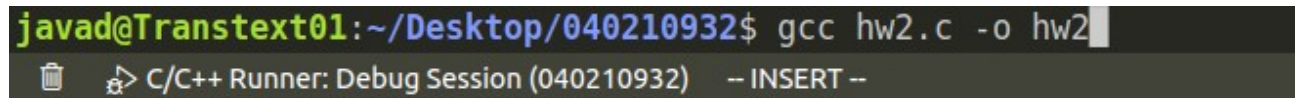
12-15 – The lines where we can see the names of the cities where we have asked to find.

- Don't forget to set both source code and input file addresses correctly. It is vital in order to run the code. My directory:

javad@Transtext01:~/Desktop/040210932\$

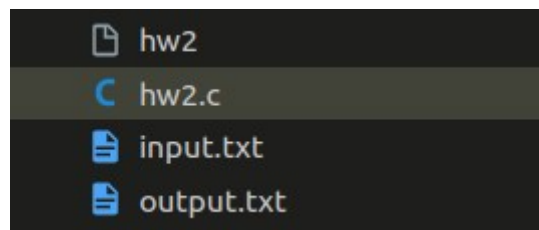
### 2.2 Run the code

I will run the code with the help of the gcc.

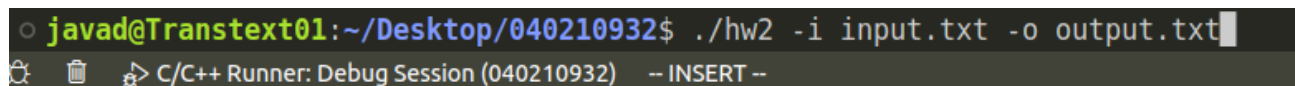


```
javad@Transtext01:~/Desktop/040210932$ gcc hw2.c -o hw2
```

After running this code a new file named “hw2” will be created. Kjj



Then I use this command to run this code :



```
javad@Transtext01:~/Desktop/040210932$ ./hw2 -i input.txt -o output.txt
```

d

Don't forget that you can set the name of output.txt whatever you want, but input file must be previously chosen.

### 2.3 Terminal Output

This code will return 2 things- One output.txt file and one terminal output(b)

```
output.txt
1 Path(Prague London): Prague -> Helsinki -> Tokyo -> NewYork -> London
2 Distance: 26065 km
3 Path(London London): London -> Tokyo -> NewYork -> London
4 Distance: 25971 km
5 Path(London Prague): Path not found
6 Distance: Path not found
7
```

A)

```
javad@Transtext01:~/Desktop/040210932$ ./hw2 -i input.txt -o output.txt
Path (Prague London): Prague -> Helsinki -> Tokyo -> NewYork -> London
Distance: 26065 km
Path (London London): London -> Tokyo -> NewYork -> London
Distance: 25971 km
Path (London Prague): Path not found
Distance: Path not found
```

B)

OUTPUT value is TRUE. The basic logic is that it is impossible to go to some cities from some cities. After we were able to solve this problem with possible possibilities, we worked more comfortably.

This document will be shared in **MY GITHUB ACCOUNT** at the end of the deadne.

Link:

<https://github.com/cavadibrahimli1/ITU-homeworks/tree/main/FInd%20the%20shortest%20path%20between%20two%20cities>