Ankara University
Computer Engineering
COM2067 LAB 1

	A	B	C	D	E	F	G	H	I	J
A	0	40	8	27	-1	-1	-1	-1	-1	-1
B	-1	0	-1	-1	6	-1	10	-1	-1	-1
C	-1	7	0	12	-1	2	-1	-1	-1	-1
D	-1	-1	-1	0	-1	9	-1	-1	-1	-1
E	-1	-1	10	-1	0	12	11	-1	-1	-1
F	-1	-1	-1	-1	-1	0	-1	13	14	-1
G	-1	-1	-1	-1	-1	-1	0	20	-1	15
H	-1	-1	-1	-1	17	-1	-1	0	-1	18
I	-1	-1	-1	19	-1	-1	-1	21	0	20
J	-1	-1	-1	-1	-1	-1	-1	-1	-1	0

In this lab, you are expected to find the shortest path between cities in the C programming language. First you have to create a two-dimensional array of integers of size 15 x 15 and get the values of this array from the user. This matrix will show the distance between cities. The value "-1" indicates that there is no road between the two cities. Then, you should get the two cities where the shortest route will be computed from the user and send these cities and the two-dimensional array to the function whose prototype is given below. Disrupting the function prototype will result in your lab not being evaluated.

int* functionFindMin(int* array, int row, int cols, char source, char destination);

When your function finds the shortest path, it should send the cities and the shortest distance between the two cities to the main function. The sent values will be printed on the screen with "-". If there is no connection between the two cities, the warning "You cannot travel between two cities" should be displayed on the screen.

1- For example, let the cities entered by the user be A and D. There are many alternative routes from A to D.

A-D 27
A-C-D 20
A-C-F-I-D 43
A-B-E-C-D 68
The shortest path is A-C-D and your output will be A-C-D 20.

2- For example, the cities entered by the user are F and J. There are many alternative routes from F to J.

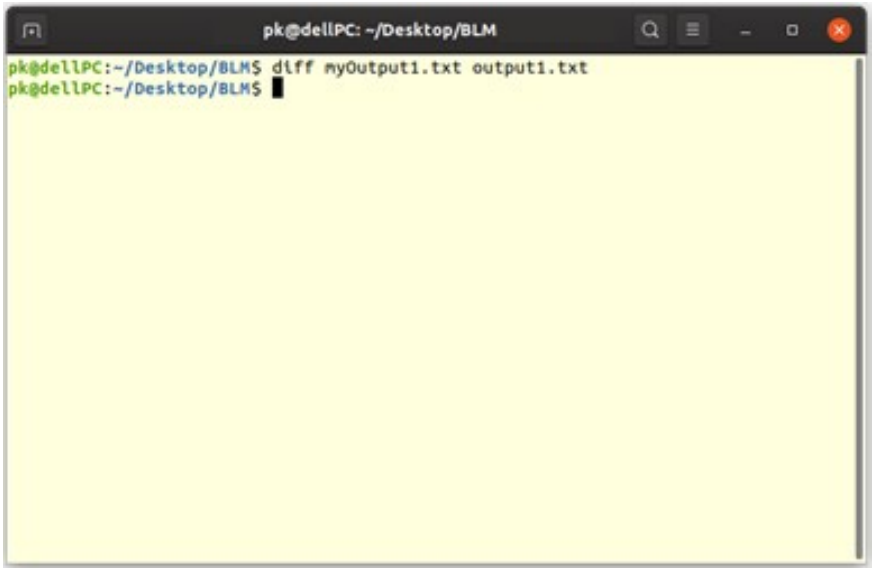
F-H-J 31
F-I-J 34
F-H-E-G-J 56
F-I-H-J 53

The shortest path is F-H-J and your output will be F-H-J 31.

There may be many alternative ways other than those given in the examples above. Only a few are given with the correct results.

Name your work as StudentID.c and upload it to the system. Make sure that your program is running in the Ubuntu environment. For the correct output format, carefully review the sample input and output files provided. You can perform the following operations to check the accuracy of your program.

1. **Compile your program using the gcc command.**
2. **Using ./a.out <input1> myOutput1.txt command, run your program with the input1.txt file and save your output to myOutput1.txt file.**
3. **Automatically compare the true output and your output using the diff myOutput1.txt output1.txt command. If there is no warning on the command prompt after entering this command, this means that your program is working correctly for these values. If you see a warning, this indicates a problem with your output.**



4. **Try commands in items 2 and 3 for other input files given to you.**
5. **Test your program for different inputs you will create yourself. Note that the input files given to you and the input files used during the evaluation may differ from each other.**