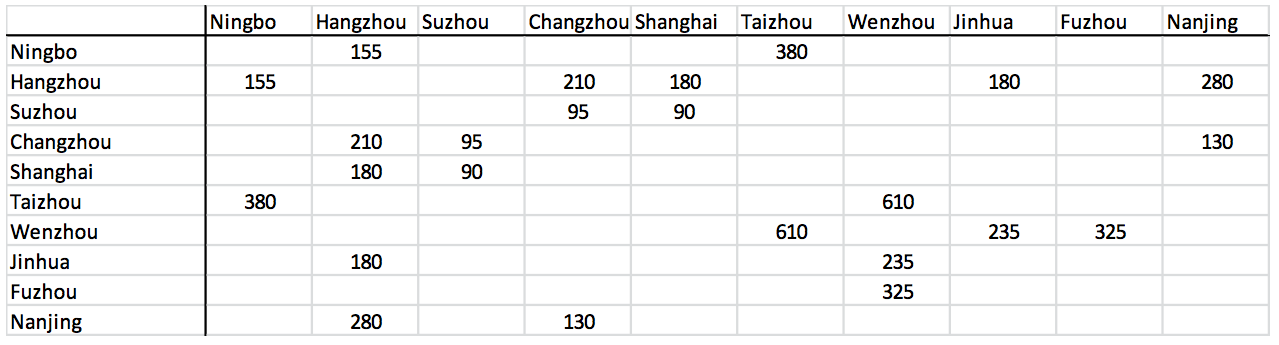
**Task**

Your task is to write a program that will check the route and price of train tickets between stations. A representation of rail connections between stations will be stored in a file that your program will need to load. The filename will be specified as a command line argument. Your program should load this file and use it to make a graph representation of the train network. It should then prompt the user to enter the names of the start and end stations, and then use some graph algorithm of your chosing to decide the shortest route between the stations, and display it to the user. It should repeatedly do this until the user wants to quit the program.

If the program is run without the correct number of command line parameters, it should exit with the error message "Invalid command line arguments. Usage: train <disances.txt>". The filename may be a relative or absolute pathname which can be understood by fopen. If the file cannot be opened, use perror to print the error message "Cannot open file." together with the operating system specific error message and exit with exit code 1.

The format of the distance file is as follows:

* Lines are separated by a single newline character '\n'.
* On each line, there are multiple cells. Each cell is separated by a comma character ','.
* Each cell may or may not have a value. Cells without values have nothing between commas or between the start/end of the line and the comma.
* Each value can contain any number of any printable ASCII characters, excluding the comma ',' and newline '\n' characters.
* Blank lines (lines with no characters) at the end of the file should be ignored.
* The first line of values in the cells are organised like a table. The first row are the names of stations. The first column are the name of stations. The top-left cell should be empty. The rest of the values of the cells represents the distance in kilometers from the station of that row to the station of that column (ie, from the station of the left to the station on the top). If the cell is empty, that represents no direct connection between those two stations. A visualisation of an example table is below:



* Files which do not match this format are invalid. In particular, if a distance cell contains something other than a positive non-zero integer or empty then it is invalid.
* In the case of invalid files, the program should print the error message "Invalid distances file." and exit with exit code 2.

Your program should load the data from this file into a graph data-structure, where the stations are vertexes, the connections are directed edges, and distances are edge weights. You will then use this data-structure to calculate the journeys required below. Your program should not have to access the file again beyond this point. Note that connections do not have to be symmetric, ie, there may be pairs of stations where you can travel from A to B but not B to A. You can assume that the graph data will be a *weakly connected directed graph*.

Once the graph has been created, you should prompt the user with the message "Start station: " and read in a string. The user can quit the program by entering nothing for the start station; the program should then exit with exit code 0. Otherwise, if that string is not the name of a station then the program should print the error message "No such station." and prompt for the start station again. All user input should be matched case-sensitively.

After the start station, you should prompt the user with the message "End station: " and read in a string. If that string is not the name of a station then the program should print the error message "No such station." and prompt for the *start* station again. If the start and end station are the same, the program should print "No journey, same start and end station." and prompt for the *start* station again.

If both station names are valid, the program should calculate the shortest journey from the start to the end station. You should chose an appropriate graph algorithm to do this efficiently to give you the data you need to display the output below. The program should print out the journey in the following format:

From S

via

C1

C2

C3

...

To E

Distance XXX km

Cost YYY RMB

where *S* and *E* are the start and end station names; *C1, C2, C3, ...* are all the intermediate station names that the journey will go through; *XXX* is the total distance of the journey; and *YYY* is the total cost of the journey (see next section). If there are no intermediate stations, then the output should display "direct" instead of "via" (and obviously not print any intermediate stations). Once this has been printed out then it should prompt for the start station again.

If there is no possible journey between the stations then the program should print "No possible journey." and prompt for the start station again.

The cost of the journey is calculated as *(the total distance multiplied by 1.2) + (the number of intermediate stations multiplied by 25)*. If the result is not a whole number then it should be rounded up to the next nearest integer.

If the program needs to exit because it cannot allocate memory, it should print the error message "Unable to allocate memory." and exit with exit code 3. If the program needs to exit for any other reason not covered in this document, print an appropriate error message and exit with exit code 4.

**Example input/output**

Given the following distances file (distances1.txt, with the file in the current directory):

,Ningbo,Hangzhou,Suzhou,Changzhou,Shanghai,Taizhou,Wenzhou,Jinhua,Fuzhou,Nanjing

Ningbo,,155,,,,380,,,,

Hangzhou,155,,,210,180,,,180,,280

Suzhou,,,,95,90,,,,,

Changzhou,,210,95,,,,,,,130

Shanghai,,180,90,,,,,,,

Taizhou,380,,,,,,610,,,

Wenzhou,,,,,,610,,235,325,

Jinhua,,180,,,,,235,,,

Fuzhou,,,,,,,325,,,

Nanjing,,280,,130,,,,,,

Running the program and just pressing return:

zlizpd3 $ ./train distances1.txt

Start station:

zlizpd3 $

Running the program:

zlizpd3 $ ./train distances1.txt

Start station: Ningbo

End station: Suzhou

From Ningbo

via

Hangzhou

Shanghai

To Suzhou

Distance 425 km

Cost 560 RMB

Start station: Ningbo

End station: Ningbo

No journey, same start and end station.

Start station: Glasgow

No such station.

Start station: Nanjing

End station: Glasgow

No such station.

Start station: 341ed admom1 q!!!!

No such station.

Start station: Wenzhou

End station: Fuzhou

From Wenzhou

direct

To Fuzhou

Distance 325 km

Cost 390 RMB

Start station:

zlizpd3 $

Running the program with an invalid file:

zlizpd3 $ ./train invalid1.txt

Invalid distances file.

zlizpd3 $

Running the program with a wrong filename:

zlizpd3 $ ./train missing.txt

Cannot open file.*<OS-specific perror message here>*

zlizpd3 $