DOCUMENTATION FOR
QUESTION 1:

## Conversion of a DFA to a minimum state DFA

*Example from dfa.csv:*

Transition Table for the DFA :

| states | On input 0 | On input 1 |
|---|---|---|
| ->A (initial state) | B | C |
| B | A | D |
| C(final) | E | F |
| D(final) | E | F |
| E(final) | E | F |
| F | F | F |

NOTE : The order of rows in the dfa.csv file has all non-final states above & final states below '*' for convenience of reading data from file

Firstly, we divide the states in two sets, namely set of non-final states & set of final states. This is nothing but the 'zero' equivalence.
And then we proceed for each of the sets to divide them further and stop once the division gives us the same result.

For the above DFA,

| 0 equivalence : {A,B,F} , {C,D,E} |
|---|

We first compare transitions on input 0 & input 1 of B with A.
On input 0, A goes to B & B goes to A. Both transition states B & A belong to the same the set. And on input 1, A goes to C & B goes to D. Both transition states C & D belong to the same set. Thus, A & B belong to the same set in 1 equivalence.
Therefore, we have {A,B} in 1 equivalence.
Now check for F with A or B. We will check with A.
On input 0, A goes to B & F goes to F. So both belong to same set. Now we check on input 1. On input 1, A goes to C & F goes to F. Now since C & F belong to different sets & there is no other set to compare with, {F} is another set in 1 equivalence.
Now check for D & C. We observe, they belong to same set. Thus, we have {C,D}.

Now we check for E & C. We observe, they too belong to the same set. Thus, we include E in the set of C. Thus, we have {C,D,E}.
Hence,

| 1 equivalence : {A,B}, {F}, {C,D,E} |
|---|

Similarly, we proceed for 2 equivalence and we get as follows,

| 2 equivalence : {A,B}, {F}, {C,D,E} |
|---|

As we observe, 1 equivalence & 2 equivalence are same we stop & these then become the states in the minimized DFA.

Transition Table for the MINIMIZED DFA :

| states | On input 0 | On input 1 |
|---|---|---|
| ->{A,B} | {A,B} | {C,D,E} |
| {F} | {F} | {F} |
| {C,D,E}(final) | {C,D,E} | {F} |

Solution approach in my program:
1.Initially, the equivalence 0 is stored in the 2D vector of vector of strings called as ‘equivalence’.
2.In each iteration, set of states in current equivalence are broken down or combined into sets and the contents of each iteration are stored in 2D vector of vector of strings called as ’temp’.
3.And the contents are then appended in another 2D vector of vector of strings called as ‘equivalence_new’.
4.After all rows are done, the contents of ‘equivalence_new’ are copied into ‘equivalence’ since now we need to iterate over this new equivalence to get the next equivalence.
5.As soon as ‘equivalence_new’ is exactly same as that of ‘equivalence’, we exit from while loop indicating that the states for the minimum state DFA are obtained.
6.Then from the transition table of the DFA, the transitions for the states of the minimized DFA are obtained and thus the transition table for the minimized DFA is obtained.