

Programming Assignment 2

How to run

```
g++ qx.cpp  
./a.out <input file> <output file>
```

Here, x can be {3, 4}

Question 3: Minimisation of DFA

- Removal of dead states
- Use the equivalence theorem and separate the states into final and non-final states
- This is followed by iteration and grouping of equivalent states and is done till the two consecutive partitions are identical
- These states obtained in the end represent the **states** of our **minimized DFA**
- **Letters** would be same as that of the input DFA
- Similarly, the **start states** and **end states** are also same as that of the input
- For the transition function, we can look at the following example to understand the process well
 - Say the **start_states** are something like [0, 1, 2], [3,4], [5,6] and let one of the transitions be 2 -a-> 6. From this, we can get the new transition of the minimized DFA as [0,1,2] -a-> [5,6]
 - The above step is done for all the states to obtain the final transition matrix

Question 4: Regex -> NFA

- Add the concatenation operator . at the required places
- Convert the given expression to a **postfix** expression for easier processing
 - Uses **Shunting Yard Algorithm**
- Use this postfix expression obtained to create NFA using **Thompson's Construction**
- During this process, we keep on splitting the regex till each edge of our NFA represents only one character – addition of these different edges is done in accordance with the operator (like union, concatenation etc.)