- Muhammad Shayan Ejaz
- FA20/BSAI/006

DFA 1:

```
class DFA1:
    initial_state = 0
    word = ''
    valid = False
    current_state = 0

    def _int_(self):
        self.word = ''
        self.initial_state = 0
        self.valid = False

    def constructor(self, w, i):
        self.word = w
        self.initial_state = i

    def transition(self, alphabet, state):
        if state == 0:
            if alphabet == 0:
                state = 1
            elif alphabet == 1:
                state = 3
            else:
                print("Invalid Character: ", alphabet)
        elif state == 1:
            if alphabet == 0:
                state = 0
            elif alphabet == 1:
                state = 2
            else:
                print("Invalid Character: ", alphabet)
```

```
        elif state == 2:
            if alphabet == 0:
                state = 3
            elif alphabet == 1:
                state = 1
            else:
                print("Invalid Character: ", alphabet)
        elif state == 3:
            if alphabet == 0:
                state = 2
            elif alphabet == 1:
                state = 0
            else:
                print("Invalid Character: ", alphabet)
        return state

    def DFA_working(self, new_word):
        self.word = new_word
        for i in self.word:
            self.current_state = self.transition(int(i), self.current_state)
        if self.current_state == 0:
            print("VALID STRING")
        else:
            print("INVALID STRING")
if __name__ == "__main__":
    print("DFA 3 STARTING")
    inputString = str(input("Enter the String: "))
    S = DFA1()
    S.DFA_working(inputString)
```

```
DFA 3 STARTING
Enter the String: 001123
Invalid Character:  2
Invalid Character:  3
VALID STRING
```

DFA 2:

```
class DFA2:
    word=' '
    initial_state=0
    current_state=0

    def _init_(self):
        self.word=' '
        self.initial_state=0
    def constructor(self,w,i):
        self.word=w
        self.initial_state=i

    def transitionfunction(self,alphabet,state):
        if state==0:
            if alphabet==0:
                state=3
            elif alphabet==1:
                state=1
            else:
                print("Invalid Character: ", alphabet)

        elif state==1:
            if alphabet==0:
                state=1
            elif alphabet==1:
                state=2
            else:
                print("Invalid Character: ", alphabet)

        elif state==2:
            if alphabet == 0:
                state = 2
            elif alphabet == 1:
                state = 2
            else:
                print("Invalid Character: ", alphabet)

        elif state==3:
```

```
            if alphabet == 0:
                state = 4
            elif alphabet == 1:
                state = 3
            else:
                print("Invalid Character: ", alphabet)

        elif state==4:
            if alphabet == 0:
                state = 4
            elif alphabet == 1:
                state = 4
            else:
                print("Invalid Character: ", alphabet)

        return state

    def DFA_working(self,neword):
        self.word = neword
        for i in self.word:
            self.current_state = self.transitionfunction(int(i), self.current_state)
        if self.current_state ==  2 or self.current_state ==  4:
            print("VALID STRING")
        else:
            print("INVALID STRING")
if __name__ == "__main__":
    print("DFA 3 STARTING")
    inputString = str(input("Enter the String: "))
    S = DFA2()
    S.DFA_working(inputString)
```

```
DFA 3 STARTING
Enter the String: 000112
Invalid Character:  2
VALID STRING
```

DFA 3:

```
class DFA3:
    word = ''
    initial_state = 0
    current_state = 0
    valid = False

    def __int__(self):
        self.word = ''
        self.initial_state = 0

    def constructor(self, w, i):
        self.word = w
        self.initial_state = i

    def transition(self, alphabet, state):
        if state == 0:
            if alphabet == 'c':
                state = 1
            elif alphabet == 'a':
                state = 2
            elif alphabet == 'b':
                state = 2
            else:
                print("INVALID CHARACTER: ", alphabet)

        elif state == 1:
            if alphabet == 'a':
                state = 3
            elif alphabet == 'b':
                state = 2
            elif alphabet == 'c':
                state = 2
            else:
                print("INVALID CHARACTER: ", alphabet)

        elif state == 2:
            if alphabet == 'a':
                state = 3
```

```
            elif alphabet == 'b':
                state = 3
            elif alphabet == 'c':
                state = 3
            else:
                print("INVALID CHARACTER: ", alphabet)

        elif state == 3:
            if alphabet == 'a':
                state = 3
            elif alphabet == 'b':
                state = 3
            elif alphabet == 'c':
                state = 3
            else:
                print("INVALID CHARACTER: ", alphabet)
        return state


    def DFA_working(self, neword):
        self.word = neword
        for i in self.word:
            self.current_state = self.transition(str(i), self.current_state)
        if self.current_state == 2:
            print("VALID STRING")
            DFA_file3 = open("DFA_file3.txt", "a")
            DFA_file3.write(inputString + "\n")
            DFA_file3.close()
        else:
            print("INVALID STRING")


if __name__ == "__main__":
    print("DFA 3 STARTING")
    inputString = str(input("Enter the String: "))
    S = DFA3()
    S.DFA_working(inputString)
```

```
 DFA 3 STARTING
```

```
Enter the String: shayan
INVALID CHARACTER:  s
INVALID CHARACTER:  h
INVALID CHARACTER:  y
INVALID CHARACTER:  n
INVALID STRING
```