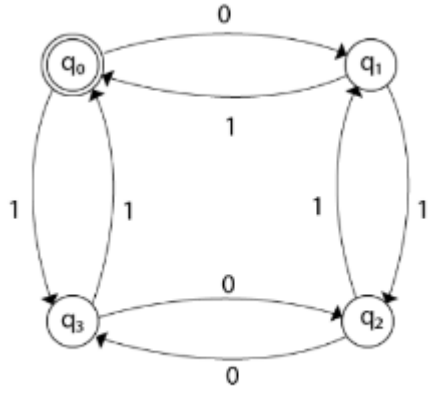
MADE BY:

AAZAIN SHAMSI

BSAI-014

DFA 1

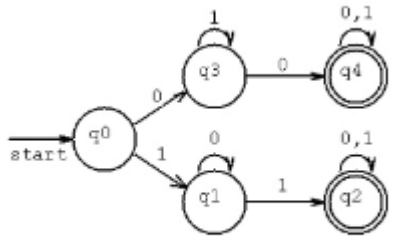



```
class DFA1:
    initial_state = 0
    word = ''
    valid = False
    current_state = 0

    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: 11
VALID STRING
```

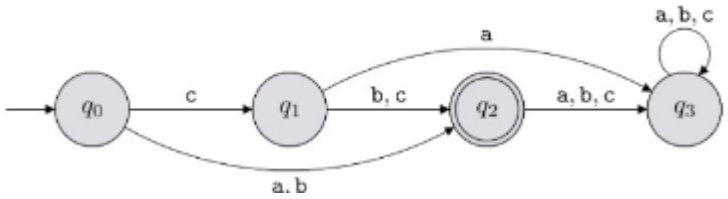
DFA 2




```
class DFA2:
    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 = 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, 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 == 2 or self.current_state==4:
            print("String is valid.")
        else:
            print("String is invalid.")
```

DFA 3




```
class DFA3:
    word=' '
    initial_state=0
    final_state=0
    current_state=0

    def _init_(self):
        self.word=' '
        self.initial_state=0
        self.final_state=0
    def constructor(self,w,i,c,f):
        self.word=w
        self.initial_state=i
        self.current_state = c
        self.final_state=f
```

```
    def transitionfunction(self,alphabet,state):
        if state==0:
            if alphabet=='a':
                state=2
            elif alphabet=='b':
                state=2
            elif alphabet=='c':
                state=1
            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.transitionfunction(self,i, self.current_state)
        if self.current_state == self.final_state:
            print("VALID STRING: ", neword,"\n")
        else:
            print("INVALID STRING: ", neword,"\n")
```

```
DFA3()
word='cc'

DFA3.constructor(DFA3,word,0,0,2)
DFA3.DFA_working(DFA3,word)
```

```
    VALID STRING:  cc
```