

```
"""
```

```
Zac Foteff, Wesley Muehlhausen  
CPSC351: 01  
Problem Set #10
```

```
Program simulates the behavior of a Turing Machine  
"""
```

```
class TM():  
    def __init__(self, w, head, q=['q1', 'q2', 'q3', 'q4', 'q5', 'qa', 'qr'],  
                  sigma=['0'],  
                  gamma=['0', ' ', 'x'],  
                  delta={('q1', '0'):(('q2', ' ', 'R'),  
                                       ('q1', ' '), ('qr', ' ', 'R'),  
                                       ('q1', 'x'), ('qr', ' ', 'R'),  
                                       ('q2', 'x'), ('q2', ' ', 'R'),  
                                       ('q2', '0'):(('q3', 'x', 'R'),  
                                                    ('q2', ' '), ('qa', ' ', 'L'),  
                                                    ('q3', ' '), ('q5', ' ', 'L'),  
                                                    ('q3', 'x'), ('q3', ' ', 'R'),  
                                                    ('q3', '0'):(('q4', ' ', 'R'),  
                                                    ('q4', ' '), ('qr', ' ', 'R'),  
                                                    ('q4', 'x'), ('q4', ' ', 'R'),  
                                                    ('q4', '0'):(('q3', 'x', 'R'),  
                                                    ('q5', ' '), ('q2', ' ', 'R'),  
                                                    ('q5', '0'):(('q5', ' ', 'L'),  
                                                    ('q5', 'x'):(('q5', ' ', 'L'))},  
                  Q0='q1',  
                  ACCEPT=['qa'],  
                  REJECT=['qr']  
    ):  
        self.q = q  
        self.sigma = sigma  
        self.gamma = gamma  
        self.delta = delta  
        self.Q0 = Q0  
        self.ACCEPT = ACCEPT  
        self.REJECT = REJECT  
        self.w = w  
        self.head = head  
  
    # R/W head manipulation methods  
    def deltaTransition(self, state, head):  
        newstate, writeSymbol, tapeDir = self.delta[state, head]  
        return newstate, writeSymbol, tapeDir  
  
    def simulate(self):  
        state = self.Q0  
        tape = list(self.w + " ")
```

```

    head = 0
    while head < len(tape):
        symbol = tape[head]
        state, writeSymbol, tapeDir = self.deltaTransition(state, symbol)

        if state in self.ACCEPT:
            print('Accept')
            break

        if state in self.REJECT:
            print("Reject")
            break

        if writeSymbol != '':
            if (writeSymbol not in self.gamma):
                print('Reject')
                break
            tape[head] = writeSymbol

        if tapeDir == 'L':
            head -= 1
        elif tapeDir == 'R':
            head += 1

def main():
    while True:
        w = input("Please input a string to test against the Turing Machine, or q to
quit: ")
        if w == "q":
            break

        if len(w) > 0:
            m = TM(w, w[0])
            m.simulate()

main()

```