```
import random
import math

def setZero(x):
        if x < 0:
                return 0
        else:
                return x

def lightCycles():
    count = 0
    temp = carsL
    carsPass = math.floor(lightTime/reactionTime)
    while carsPass < temp:
        if carsPass >= temp:
            break
        else:
            temp = temp - carsPass
            count = count + 1
    return count

def cycleWait():
    count = 0
    temp = carsRightS + carsML
    carsPass = math.floor(lightTime/reactionTime)
    while carsPass < temp:
        if carsPass >= temp:
            break
        else:
            temp = temp - carsPass
            count = count + 1
    return count

def oncTraffic():
    if (carsOppS * reactionTime) + (carsL * reactionTime) + carsL < lightTime:
        return True
    else:
        return False

############################################
def SLset1():
    if oncTraffic():
        SLtime = (carsOppS * reactionTime) + (carsL * reactionTime) + carsL
    else:
        SLtime = ((lightCycles() * 5) + 1) * (lightTime + lightDelay) + (carsLcur * reactionTime)
    return int(SLtime)

def SLset2():
    if (lightCycles() > 0) and oncTraffic():
        SLtime = (lightCycles() * 4) * (lightTime + lightDelay) + (carsOppS * reactionTime) + ((carsL - (setZero(math.floor(lightTime / reactionTime)))) * reactionTime) + carsL + 100000000000
    else:
        SLtime = (lightCycles() * 5) * (lightTime + lightDelay) + (carsLcur * reactionTime)
    return int(SLtime)

def SLset3():
    if oncTraffic():
        SLtime = (3 * (lightTime + lightDelay)) + (carsOppS * reactionTime) + (carsL * reactionTime) + carsL
    else:
        SLtime = ((lightCycles() * 5) + 4) * (lightTime + lightDelay) + (carsLcur * reactionTime)
    return int(SLtime)

def SLset4():
    if oncTraffic():
        SLtime = (2 * (lightTime + lightDelay)) + (carsOppS * reactionTime) + (carsL * reactionTime) + carsL
    else:
        SLtime = ((lightCycles() * 5) + 3) * (lightTime + lightDelay) + (carsL * reactionTime)
    return int(SLtime)

def SLset5():
    if oncTraffic():
        SLtime = (lightTime + lightDelay) + (carsOppS * reactionTime) + (carsL * reactionTime) + carsL
    else:
        SLtime = ((lightCycles() * 5) + 2) * (lightTime + lightDelay) + (carsLcur * reactionTime)
    return int(SLtime)
```

```
#############################################################
#########################
def MLset1():
   MLtime = (3 * lightTime) + ((cycleWait() * 5) * (lightTime + lightDelay)) +
(carsMLcur * reactionTime)
   return int(MLtime)

def MLset2():
   MLtime = (2 * lightTime) + ((cycleWait() * 5) * (lightTime + lightDelay)) +
(carsMLcur * reactionTime)
   return int(MLtime)

def MLset3():
   MLtime = (1 * lightTime) + ((cycleWait() * 5) * (lightTime + lightDelay)) +
(carsMLcur * reactionTime)
   return int(MLtime)

def MLset4():
   if (((carsML * (Rtime + Utime)) + (carsR * Rtime) + (reactionTime * (carsML +
carsRightS))) < lightTime):
      MLtime = ((carsML * (Rtime + Utime)) + (carsR * Rtime) + (reactionTime *
(carsML + carsRightS)))
   else:
      MLtime = (((cycleWait() + 1)* 5) * (lightTime + lightDelay)) + (carsMLcur *
reactionTime)
   return int(MLtime)

def MLset5():
   MLtime = (4 * lightTime) + ((cycleWait() * 5) * (lightTime + lightDelay)) +
(carsMLcur * reactionTime)
   return int(MLtime)

############################################
carsS = random.randint(0,30)
carsL = setZero(random.randint((carsS - 10),(carsS + 10)))
carsR = setZero(random.randint((carsS - 10),(carsS + 10)))

carsOppS = setZero(random.randint((carsS - 10),(carsS + 10)))
carsOppL = setZero(random.randint((carsOppS - 10),(carsOppS + 10)))
carsOppR = setZero(random.randint((carsOppS - 10),(carsOppS + 10)))

carsLeftS = setZero(random.randint((carsS - 10),(carsS + 10)))
```

```
carsLeftL = setZero(random.randint((carsLeftS - 10),(carsLeftS + 10)))
carsLeftR = setZero(random.randint((carsLeftS - 10),(carsLeftS + 10)))

carsRightS = setZero(random.randint((carsS - 10),(carsS + 10)))
carsRightL = setZero(random.randint((carsRightS - 10),(carsRightS + 10)))
carsRightR = setZero(random.randint((carsRightS - 10),(carsRightS + 10)))

totalCars = carsS + carsL + carsR + carsOppS + carsOppL + carsOppR +
carsLeftS + carsLeftL + carsLeftR + carsRightS + carsRightL + carsRightR

reactionTime = round(random.uniform(1,3),2)
Rtime = round(random.uniform(1,5),2)
Utime = round(random.uniform(1,7),2)
lightTime = random.randint(10, 60)
lightDelay = round(random.uniform(0.5,3),2)

carsYieldPass = setZero(math.floor((lightTime - (carsOppS * reactionTime)) /
reactionTime))
carsLcur = setZero(carsL - ((lightCycles() + 1) * carsYieldPass))

carsML = random.randint(0,carsR)
carsPassML = setZero(math.floor((lightTime / reactionTime)))
carsMLcur = setZero((carsRightS + carsML) - (cycleWait() * carsPassML))

###########################################
set = random.randint(1,5)
if set == 1:
        print(SLset1())
        print(MLset1())
        if (SLset1() < MLset1()):
                print("Standard Left")
        else:
                print("Michigan Left")
if set == 2:
        print(SLset2())
        print(MLset2())
        if SLset2() < MLset2():
                print("Standard Left")
        else:
                print("Michigan Left")
if set == 3:
        print(SLset3())
        print(MLset3())
```

```
    if SLset3() < MLset3():
        print("Standard Left")
    else:
        print("Michigan Left")
if set == 4:
    print(SLset4())
    print(MLset4())
    if SLset4() < MLset4():
        print("Standard Left")
    else:
        print("Michigan Left")
if set == 5:
    print(SLset5())
    print(MLset5())
    if SLset5() < MLset5():
        print("Standard Left")
    else:
        print("Michigan Left")


print(set)
print(totalCars)
print(lightTime)
```