```
#!/usr/bin/python
import re
from Cache import Cache
import sys
regex = re.compile('[0-9]+')

'''
      A simple cache simulator developed by Mohammadreza Baharani
      This code is developed for Computer Architecture subject
      University of North Carolina, Charlotte, USA
'''

def readFile(fileAddr):
    instructions=[]
    dataAddr=[]
    with open(fileAddr) as f:
        lines = f.readlines()
        for line in lines:
            data = line.split(' ')
            type = int(data[0],  10)
            addr = int(data[1],  16)
            if(type == 2):
                instructions.append(addr)
            elif(type == 0 or type == 1):
                dataAddr.append(addr)
            else:
                print(data[0])
    return [instructions, dataAddr]

def simulateInstructionCache(inst,  ch):
    i=0
    length = len(inst)
    for addr in inst:
        ch.read(addr)
        i = i+1
        sys.stdout.write(ch.name + " simulation completed: 
{0:.2f}%\r".format(float(i)*100/length))
        sys.stdout.flush()
    printResult(ch, length)

def simulateDataCache(addrData,  ch):
    i=0
    length = len(addrData)
    for addr in addrData:
        ch.read(addr)
        i = i+1
        sys.stdout.write(ch.name + " simulation completed: 
{0:.2f}%\r".format(float(i)*100/length))
        sys.stdout.flush()
    printResult(ch, length)
```

```
def printResult(ch, totalAddr):
    print
    print("Simulation is finished")
    print("\tNumber of  addresses: " + str(totalAddr))
    print("\tResult for " + ch.name +":")
    print("\tTotal     : " + str(ch.access))
    print("\tMisses     : " + str(ch.miss))
    print("\tHit     : " + str(ch.access - ch.miss))
    print("\tHit Rate : {0:.5}".format(float(ch.access -
ch.miss)*100/ch.access))


if __name__ =='__main__':
    if(len(sys.argv) < 4):
        print(sys.argv[0] + " fileTrace cacheSize(k) blockSize
setNumber")
        quit()

    filePath = sys.argv[1]
    cacheSize = 16 * int(sys.argv[2]) * 1024
    blockSize = int(sys.argv[3])
    setNumber = int(sys.argv[4])
    [inst, dataAdr] = readFile(filePath)
    l1= Cache(32, 'l1_icache',  cacheSize,  blockSize,
setNumber)
    l1_d= Cache(32, 'l1_dcache',  cacheSize,  blockSize,
setNumber)
    l1.construct()
    l1_d.construct()
    simulateDataCache(dataAdr,  l1_d);
    simulateInstructionCache(inst,  l1);
```