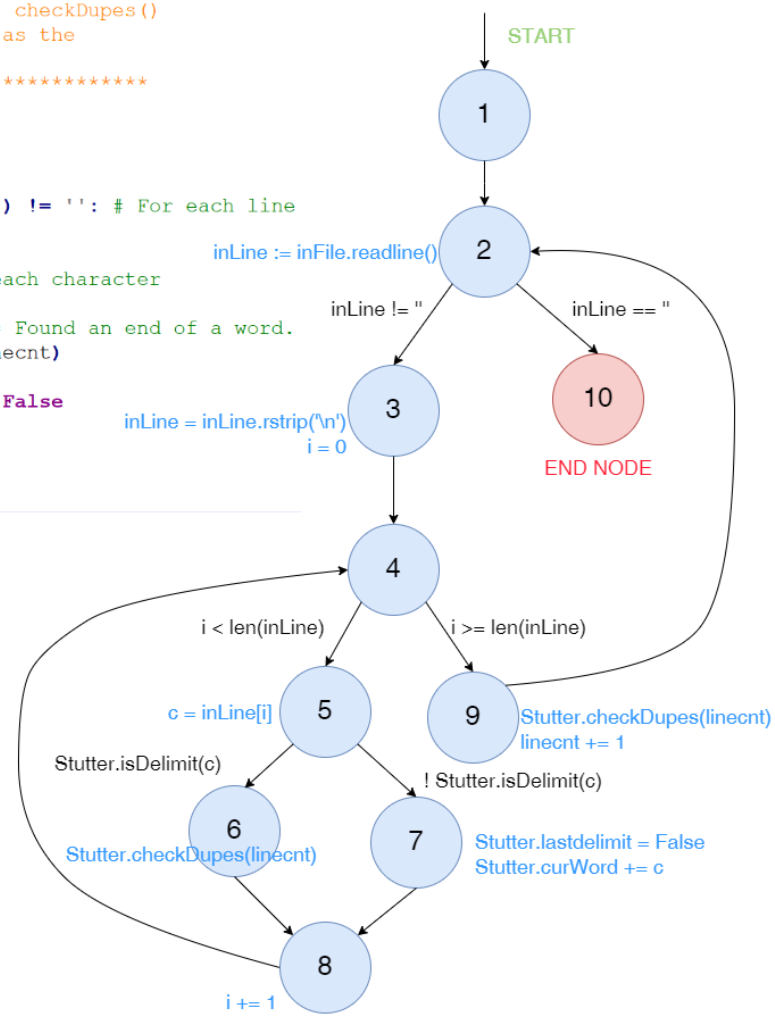
Homework 3: Test for Stutter

Use "Stutter.py" for questions a-d. (~~A compilable version is available in https://cs.gmu.edu/~offutt/softwaretest/java/Stutter.java. A line-numbered version is in https://cs.gmu.edu/~offutt/softwaretest/java/Stutter.num.~~)

(a) Draw control flow graphs for all the methods in "Stutter.py".

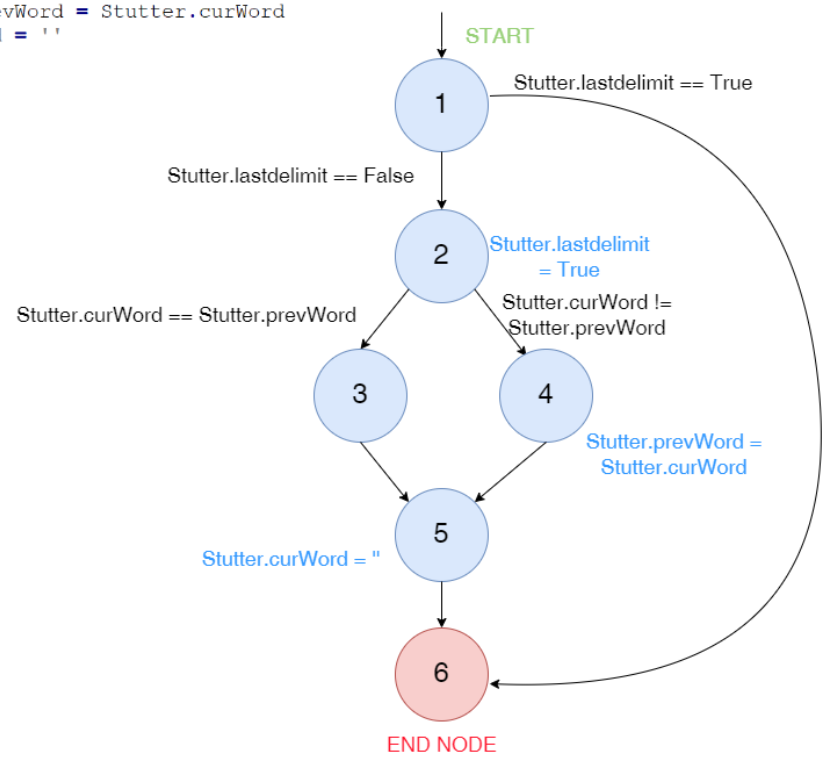
stuts(inFile):

```
'''
//************************************************
// Stut() reads all lines in the input stream, and
// finds words. Words are defined as being surrounded
// by delimiters as defined in the delimits[] array.
// Every time an end of word is found, checkDupes()
// is called to see if it is the same as the
// previous word.
//************************************************
'''
@staticmethod
def stut(inFile):
    linecnt = 1
    while (inLine := inFile.readline()) != '': # For each line
        inLine = inLine.rstrip('\n')
        i = 0
        while i < len(inLine): # for each character
            c = inLine[i]
            if Stutter.isDelimit(c): # Found an end of a word.
                Stutter.checkDupes(linecnt)
            else:
                Stutter.lastdelimit = False
                Stutter.curWord += c
            i += 1
        Stutter.checkDupes(linecnt)
        linecnt += 1
```



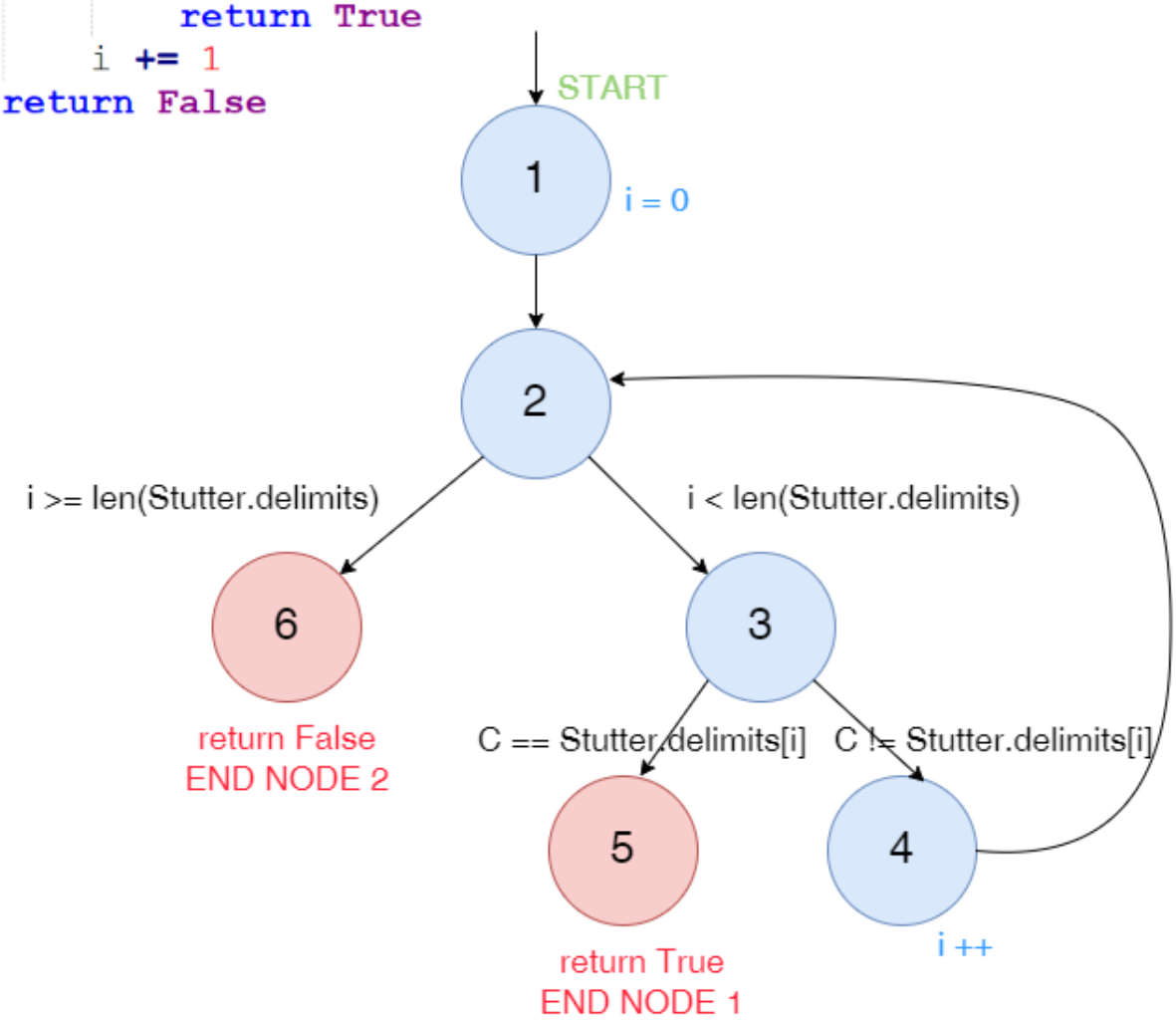
checkDupes(line):

```
'''
//*********************************************
// checkDupes() checks to see if the globally defined
// curWord is the same as prevWord and prints a message
// if they are the same.
//*********************************************
'''
@staticmethod
def checkDupes(line):
    if Stutter.lastdelimit:
        return # already checked, keep skipping
    Stutter.lastdelimit = True
    if Stutter.curWord == Stutter.prevWord:
        print('Repeated word on line ', line, ': ', Stutter.prevWord, ' ', Stutter.curWord, sep='')
    else:
        Stutter.prevWord = Stutter.curWord
    Stutter.curWord = ''
```



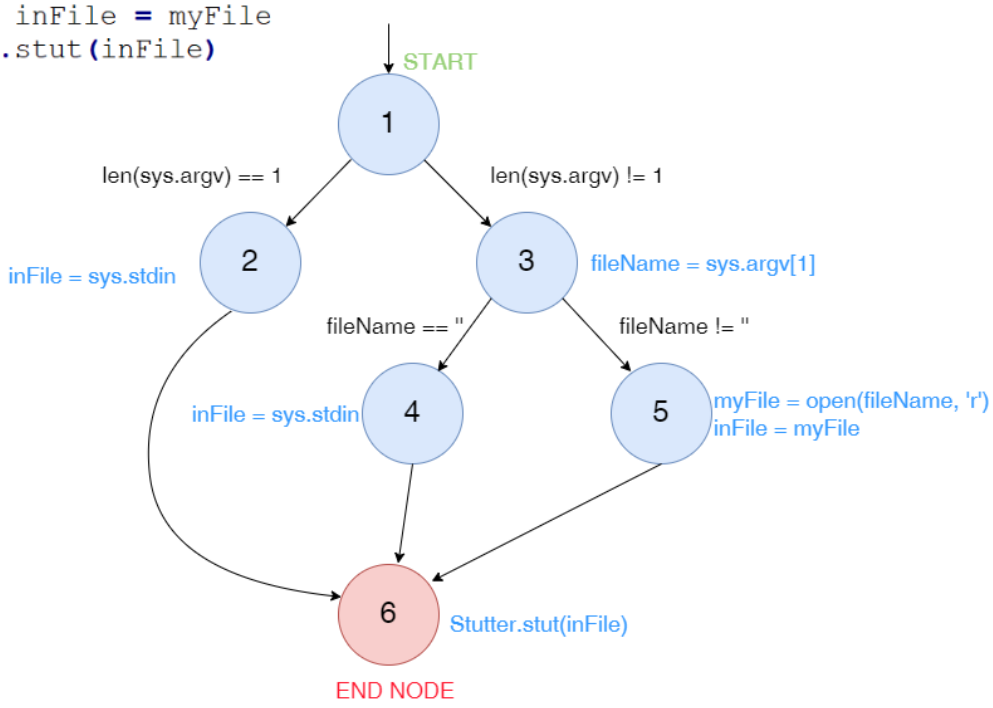
isDelimit(c):

```
'''
//*************************************************
// Checks to see if a character is a delimiter.
//*************************************************
'''
@staticmethod
def isDelimit(C):
    i = 0
    while i < len(Stutter.delimits):
        if C == Stutter.delimits[i]:
            return True
        i += 1
    return False
```



main():

```
'''
//************************************************
// main parses the arguments, decides if stdin
// or a file name, and calls Stut().
//************************************************
'''
def main():
    if len(sys.argv) == 1: # no file, use stdin
        inFile = sys.stdin
    else:
        fileName = sys.argv[1]
        if fileName == '': # no file name, use stdin
            inFile = sys.stdin
        else: # file name, open the file.
            myFile = open(fileName, 'r')
            inFile = myFile
    Stutter.stut(inFile)
```



(b) List all the call sites.

Caller	Callee	Callsite
main()	stut(inFile)	line number 103
stut(inFile)	checkDupes(line)	line number 47
stut(inFile)	checkDupes(line)	line number 52
stut(inFile)	isDelimit(C)	line number 46

(c) List all coupling du-pairs for each call site.

Last-def			First-use			ID
Method	Variable name	Line number	Method	Veriable name	Line number	
main()	inFile	95	stut(inFile)	inFile	41	1
main()	inFile	99	stut(inFile)	inFile	41	2
main()	inFile	102	stut(inFile)	inFile	41	3
stut(inFile)	linecnt	40	checkDupes(line)	line	68	4
stut(inFile)	linecnt	49	checkDupes(line)	line	68	5
stut(inFile)	c	46	isDelimit(C)	C	82	6
isDelimit(C)		83	stut(inFile)		46	7
isDelimit(C)		85	stut(inFile)		46	8

(d) Create test data to satisfy All-Coupling Uses Coverage for "Stutter.py ". (Inf ormally, to cover all coupling du-pairs in (c).)

ID	Filename	Filename source	sys.argv	inFile	inFile source
1			1	"SoftwareTesting"	sys.stdin
2	NULL	sys.argv[1]	2	"SoftwareTesting"	sys.stdin
3	"SWT.txt"	sys.argv[1]	2	"SoftwareTesting"	open(fileName, 'r')
4	"SWT.txt"	sys.argv[1]	2	"\t"	open(fileName, 'r')
5	"SWT.txt"	sys.argv[1]	2	"SoftwareTesting"	open(fileName, 'r')
6	"SWT.txt"	sys.argv[1]	2	"SoftwareTesting"	open(fileName, 'r')
7	"SWT.txt"	sys.argv[1]	2	"\t"	open(fileName, 'r')
8	"SWT.txt"	sys.argv[1]	2	"SoftwareTesting"	open(fileName, 'r')

> ...

> Then we make sure that every def reaches all possible uses

> All-uses coverage (AUC): For each set of du-paths to uses S = du (ni, nj, v), TR c ontains at least one path d in S.

> ...

> From Ch07-1 "Data Flow Test Criteria"

Hint: Please refer to ch07-4.