**ASSIGNMENT 5**

**SYSTEM PROGRAMMING LAB**

**202100102**

**MERWIN PINTO**

**ROLL NO 1**

**DIV E**

**CODE:**

import re

default = re.compile(r'&?(\w+)(=)?(\w+)?')

parameter = re.compile(r'\(P,(\d+)\)')

class macroPass2:

    def \_\_init\_\_(self):

*self*.MNT = {}

*self*.MDT\_ARRAY = []

*self*.KPDT = {}

*self*.PNTAB = {}

*self*.ICFile = open('IC.txt',mode='r')

*self*.MDT\_ARRAYFile = open('MDT.txt',mode='r')

*self*.MNTFile = open('MNT.txt',mode='r')

*self*.KPDTFile = open('KPDT.txt',mode='r')

*self*.PNTABFile = open('PNT.txt',mode='r')

*self*.output = []

*self*.outputFile = open('EXPANSION.txt',mode='w')

    def readMDT\_ARRAY(self):

*self*.MDT\_ARRAY.append([])

        for line in *self*.MDT\_ARRAYFile.readlines():

            line = line.strip('\n')

            line = line.split('\t')

*self*.MDT\_ARRAY.append(line[:-1])

    def readMNT(self):

        skipFirstLine = False

        for line in *self*.MNTFile.readlines():

            line = line.strip('\n')

            line = line.split('\t')

            if skipFirstLine == False:

                skipFirstLine = True

                continue

            else:

*self*.MNT[line[0]] = line[1:]

    def readKPDT(self):

        lines = *self*.KPDTFile.readlines()

        for macroName,value in *self*.MNT.items():

            numOfKeywordParam = int(value[1])

            kpdtp = int(value[3])

*self*.KPDT[macroName] = {}

            for i in range(kpdtp,kpdtp + numOfKeywordParam):

                line = lines[i-1].strip('\n').split('\t')

*self*.KPDT[macroName][line[0]] = line[1]

    def readPNTAB(self):

        for line in *self*.PNTABFile.readlines():

            line = line.strip('\n')

            line = line.split('\t')

*self*.PNTAB[line[0]] = line[1:]

    def createAPTAB(self,macroName):

        APTAB = []

        APTAB.append(*self*.PNTAB[macroName])

        APTAB.append([])

        APTAB.append([])

        keywordParamDict = *self*.KPDT[macroName]

        for param in APTAB[0]:

            if param in keywordParamDict:

                APTAB[1].append(keywordParamDict[param])

                if keywordParamDict[param] == "----":

                    APTAB[2].append(None)

                else:

                    APTAB[2].append(keywordParamDict[param])

            else:

                APTAB[1].append(None)

                APTAB[2].append(None)

        return APTAB

    def printAPTAB(self,APTAB):

        print("APTAB:")

        for i in range(len(APTAB[2])):

            print(i+1,APTAB[2][i],sep="\t")

        print()

    def tupleToParam(self,line:list,APTAB:list) -> list:

        result = []

        for part in line:

            find = parameter.search(part)

            if find == None:

                result.append(part)

            else:

                idx = int(find.group(1)) - 1

                result.append(APTAB[2][idx])

        return result

    def parseFile(self):

        for line in *self*.ICFile.readlines():

            line = line.strip('\n')

            line = line.split('\t')

            part\_1 = line[0]

            if part\_1 not in *self*.MNT:

*self*.output.append(line)

            else:

*#Pass actual parameters*

                part\_2 = line[1].split(', ')

                APTAB = *self*.createAPTAB(part\_1)

                for param in range(len(part\_2)):

                    find = default.search(part\_2[param])

                    if find.group(2) == None:

                        APTAB[2][param] = find.group(1)

                    else:

                        idx = APTAB[0].index(find.group(1))

                        APTAB[2][idx] = find.group(3)

*#write macro definition with actual parameters*

                mdtp = int(*self*.MNT[part\_1][2])

                print(\*line,sep="\t")

*self*.printAPTAB(APTAB)

                for macroDefLine in *self*.MDT\_ARRAY[mdtp:]:

                    if macroDefLine[0] == "MEND":

                        break

                    else:

*self*.output.append(*self*.tupleToParam(macroDefLine,APTAB))

                        print(\**self*.tupleToParam(macroDefLine,APTAB),sep="\t")

                print('\n')

*self*.writeOutputFile()

*self*.KPDTFile.close()

*self*.MDT\_ARRAYFile.close()

*self*.MNTFile.close()

*self*.PNTABFile.close()

    def writeOutputFile(self):

        for value in *self*.output:

            line = "\t".join(value)

            line += "\n"

*self*.outputFile.write(line)

*self*.outputFile.close

pass2 = macroPass2()

pass2.readMDT\_ARRAY()

pass2.readMNT()

pass2.readKPDT()

pass2.readPNTAB()

pass2.parseFile()

**OUTPUT**



