***ASSIGNMENT 2 WEEK-1***

***NAME-Pranav Sarda***

***GRNO-21810773***

***ROLLNO-321047***

***BATCH-A2***

**CODE :-**

import re

source\_file = open('1.txt','r')

sent\_tokens = source\_file.readlines()

def pass1():

    macro\_name\_table, arg\_list\_arr, alas, macro\_def\_table = ([] for i in range(4))

    mdt\_index, mnt\_index, ala\_index = (0 for i in range(3))

    intermediate\_code = ""

    #print (sent\_tokens)

    for sent in sent\_tokens:

        mdt\_entry = {'Index': 0, 'Definition': "null"}

        mnt\_entry = {'Index': 0, 'Macro Name': "null", 'MDT Index': 0}

        ala\_entry = {'Index': 0, 'Argument': "null", 'Value': '-'}

        sent = sent.strip()

        sent = sent.replace(", ", " ")

        if(';' in sent):

            sent = re.sub(';.\*', '', sent)                      ##will substitute blank whereever '; or . or \*' is occured

        word\_tokens = re.findall(r"[\S]+", sent)

        if("MACRO" in sent):

            flag = 1

            continue

        if flag == 0:

            intermediate\_code += sent+"\n"                              ##for intermedite code

        if flag >= 1:                                           ##This will run when macro is encountered

            # sent = sent.replace("\t", " ")

            mdt\_index += 1

            mdt\_entry = {'Index': mdt\_index, 'Definition': sent}

            macro\_def\_table.append(mdt\_entry)                   ##appending the mdt entry into the list

            if flag == 1:

                macro\_name = word\_tokens[0]

                mnt\_index += 1

                operands = word\_tokens[1].split(',')

                mnt\_entry = {'Index': mnt\_index, 'Macro Name': macro\_name, 'MDT Index': mdt\_index}

                macro\_name\_table.append(mnt\_entry)              ##appending the mnt entry into the list

                for operand in operands:

                    ala\_index += 1

                    ala\_entry = {'Index': ala\_index, 'Argument': operand, 'Value': '-'}

                    alas.append(ala\_entry)

                arg\_list\_arr.append(alas)                       ##appending the ala into the list

            flag += 1

        if "MEND" in sent:

            flag = 0

            alas = []

            ala\_index = 0

    print("\nIntermediate code after pass 1 :- \n")

    print(intermediate\_code)                                            ##Printing the intermedite code

    #Output Code Pass 1

    print("\n\nMacro Name Table :- \n")

    print('{:10}'.format("Index"), '{:20}'.format("Macro Name"), '{:20}'.format("MDT Index"))

    for x in macro\_name\_table:

        print('{:10}'.format(str(x["Index"])), '{:20}'.format(x["Macro Name"]), '{:20}'.format(str(x["MDT Index"])))

    for alas in arg\_list\_arr:

        print("\n\nArgument List Array for "+macro\_name\_table[arg\_list\_arr.index(alas)]["Macro Name"]+":- \n")

        print('{:10}'.format("Index"), '{:20}'.format("Argument"), '{:20}'.format("Value"))

        for x in alas:

            print('{:10}'.format(str(x["Index"])), '{:20}'.format(x["Argument"]), '{:20}'.format(x["Value"]))

    print("\n\nMacro Definition Table :- \n")

    print('{:10}'.format("Index"), '{:20}'.format("Definiton"))

    for x in macro\_def\_table:

        print('{:10}'.format(str(x["Index"])), '{:20}'.format(x["Definition"]))

    pass2(intermediate\_code, macro\_name\_table, macro\_def\_table, arg\_list\_arr)

def pass2(intermediate\_code, macro\_name\_table, macro\_def\_table, arg\_list\_arr):

    final\_code = ""

    sent\_tokens = intermediate\_code.split('\n')         ##spliting the intermediate code of pass1 into tokens

    for sent in sent\_tokens:

        # sent = sent.replace('\t', " ")

        word\_tokens = re.findall(r"[\S]+", sent)

        flag = -1

        for word in word\_tokens:

            mnt\_entries = [mnt for mnt in macro\_name\_table if mnt["Macro Name"] == word]

                                                            ##checks whether the word is in macro\_name\_table

            if len(mnt\_entries)>0:

                mnt\_entry = mnt\_entries[0]

                #print(mnt\_entry)

                flag = 1

                continue

            if flag == 1:

                operands = word

                ops = operands.split(',')                   ##operands of the macro call

                for ala\_entry in arg\_list\_arr[mnt\_entry["Index"]-1]:

                    if ala\_entry["Index"] <= len(ops):

                        ala\_entry["Value"] = ops[ala\_entry["Index"]-1]

                for mdt\_entry in macro\_def\_table:

                    if mdt\_entry["Index"] == mnt\_entry["MDT Index"]:    ##checks if mdt index matches in macro\_def\_table

                        flag = 2

                        continue

                    if mdt\_entry["Index"] >= mnt\_entry["MDT Index"] and 'MEND' in mdt\_entry["Definition"]:

                        flag = 1

                        break

                    if flag == 2:

                        sent = mdt\_entry["Definition"]                      ##defination of macro will go in sent

                        for ala\_entry in arg\_list\_arr[mnt\_entry["Index"]-1]:

                            if ala\_entry["Argument"] in mdt\_entry["Definition"]:

                                sent = sent.replace(ala\_entry["Argument"], ala\_entry["Value"]) ##replace arguments with operands

                        final\_code += sent +'\n'

                flag = 0

        if flag == -1:

            final\_code += sent +'\n'

    #Output pass 2

    print("\n\nArgument List Arrays for Pass 2 :- ")

    for alas in arg\_list\_arr:

        print("\nArgument List Array for "+macro\_name\_table[arg\_list\_arr.index(alas)]["Macro Name"]+":- \n")

        print('{:10}'.format("Index"), '{:20}'.format("Argument"))

        for x in alas:

            print('{:10}'.format(str(x["Index"])), '{:20}'.format(x["Value"]))

    print("\nOutput Code after second pass :- \n")

    print(final\_code)

pass1()

**OUTPUT :-**









