**NAME- Kundan Kabra**

**GRNO-21810729**

**ROLLNO-321023**

**BATCH-A1**

**CODE :-**

def change(m,temp):

for im in range(m,len(MDT)):

if MDT[im]== 'MEND\n':

return im,im

split = MDT[im].split()

for jm in range(len(split)):

if(split[jm].\_\_contains\_\_(temp[0])):

MDT[im] = MDT[im].replace(split[jm],temp[1])

#PASS 1 STARTED

import copy

import sys

from tabulate import tabulate

from termcolor import colored

import subprocess

file = open('macro3.txt','r', encoding='utf-8')

Lines = []

MNT = {}

MDT = []

sp = []

left = []

m=0

mntpointer = 0

put = 0

inmacro =0

function = {}

for each in file:

each = each.replace("&REG","&amp;REG")

each = each.replace(","," ")

sp.append(each.split())

Lines.append(each)

for i in range(len(Lines)):

if put == 1:

MDT.append(Lines[i])

mntpointer+=1

for j in range(len(sp[i])):

if(sp[i][j]=="MACRO"):

inmacro = 1

if len(sp[i]) == 1 :

MNT[sp[i+1][0]] = mntpointer

put = 1

else :

MNT[sp[i][j+1]] = mntpointer

MDT.append(Lines[i].replace("MACRO",""))

mntpointer+=1

put = 1

elif sp[i][j] == "MEND":

inmacro = 0

put = 0

if inmacro == 0:

left.append(Lines[i])

for l in left:

if 'MEND\n' in left:

left.remove('MEND\n')

for i,j in MNT.items():

function[i]=MDT[j].replace(i,"").split()

for k in range(len(function[i])):

if function[i][k].\_\_contains\_\_("="):

temp = function[i][k].split("=")

function[i][k] = function[i][k].replace(function[i][k],temp[1])

function1 = copy.deepcopy(function)

for i in range(len(left)):

split = left[i].split()

for j in range(len(split)):

if split[j] in MNT:

#print(split[j])

temp =left[i].replace(split[j],"").split()

for k in range (len(temp)):

if temp[k].\_\_contains\_\_("="):

temp1 = temp[k].split("=")

temp[k] = temp[k].replace(temp[k],temp1[1])

function1[split[j]][k]=temp[k]

res = []

for key, val in function.items():

res.append(val)

res = res[0] + res[1]

res = list(dict.fromkeys(res))

dou = 0

ijk = 0

n=0

#print(res)

for i in range(len(MDT)):

split = MDT[i].split()

#print(split)

for j in range(len(split)):

if MDT[i] != 'MEND\n':

m = ijk

else:

m=i+1

ijk = i+1

if split[j].\_\_contains\_\_("="):

temp = split[j].split("=")

MDT[i] = MDT[i].replace(split[j],temp[1])

m,n = change(m,temp)

for i in range(len(MNT)):

if len(MNT) > i+1 :

gotdata = MNT[list(MNT)[i+1]]

else:

gotdata = 'null'

if gotdata == 'null':

for j in range(MNT[list(MNT)[i]] , len(MDT) ):

split = MDT[j].split()

for k in range(len(split)):

if split[k] in res:

MDT[j]=MDT[j].replace(split[k],"#" +str(function[list(MNT)[i]].index(split[k])))

#print(MDT[j])

else:

for j in range(MNT[list(MNT)[i]],gotdata):

split = MDT[j].split()

for k in range(len(split)):

if split[k] in res:

MDT[j]=MDT[j].replace(split[k],"#" +str(function[list(MNT)[i]].index(split[k])))

#print(MDT[j])

mdt1 = []

ins = 0

def putinmdt(tempmdtline,tempeach,tempi):

mdt1.pop()

parameter = tempmdtline.replace(tempeach,"").split()

current\_parameter = MDT[MNT[tempeach]].replace(tempeach,"").split()

for j in range(MNT[tempeach]+1 ,len(MDT)):

if MDT[j]== 'MEND\n':

break

else:

mdt1.append(MDT[j])

spl = mdt1[len(mdt1)-1].split()

for k in range(len(spl)):

if spl[k] in current\_parameter:

mdt1[len(mdt1)-1]=mdt1[len(mdt1)-1].replace(spl[k],parameter[current\_parameter.index(spl[k])])

temp=[]

for i in MNT.keys():

temp.append(i)

for i in range(len(MDT)):

mdt1.append(MDT[i])

ins+=1

for each in temp :

if MDT[i].\_\_contains\_\_(each):

if i != MNT[each]:

#print(MDT[i])

putinmdt(MDT[i],each,i)

print()

MNT1 = {}

temp = []

for k in MNT.keys():

temp.append(k)

MNT1[k]= [len(MDT[MNT[k]].split()) - 1,MNT[k]]

print(" {}".format(colored("MNT" ,'red')))

headers = ["Name of Macro","Number of parameters","Starting Index"]

print(tabulate([[k,] + v for k,v in MNT1.items()],headers=headers,tablefmt="fancy\_grid"))

MDTSTRUCTUREDLIST = []

i=0

for each in MDT:

each1 = each.split()

temp = each.replace(each1[0],"")

MDTSTRUCTUREDLIST.insert(i,[each1[0],temp])

i+=1

headers = [colored("MDT",'red'),""]

print(tabulate(MDTSTRUCTUREDLIST,headers=headers,tablefmt="fancy\_grid",showindex=True))

file1 = open("Intermidiate code.txt","+w")

print(" {}".format(colored("Intermidiate code","red")))

for each in left:

print(" {}".format(each))

file1.write(each)

file1.close()

file.close()

**OUTPUT :-**

