**NAME- Kundan Kabra**

**GRNO-21810729**

**ROLLNO-321023**

**BATCH-A1**

**CODE :-**

mdt1 = []

ins = 0

def putinmdt(tempmdtline,tempeach,tempi): ##Function to put the calls made inside the macro in MDT

mdt1.pop()

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

for par in range(len(parameter)):

listtt.append([tempeach,parameter[par],"#" + str(par)])

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])])

def change(m,temp): ##Function to Handle the = for the parameters

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])

import copy

from collections import Iterable

from termcolor import colored

from tabulate import tabulate

file = open('macro1.txt','r', encoding='utf-8')

Lines = []

MNT = {}

MDT = []

sp = []

listtt = []

left = []

colour\_list = ['grey','green','yellow','blue','red','magenta','cyan','white']

dictoo = {}

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])

## TO PUT THE PARAMETERS MADE IN IC IN A DIFFERENT ALA

res = []

for key, val in function.items(): ## TO FLATTEN the nested lists

res.append(val)

def flatten(lis):

for item in lis:

if isinstance(item, Iterable) and not isinstance(item, str):

for x in flatten(item):

yield x

else:

yield item

res = list(flatten(res))

dou = 0

ijk = 0

n=0

for i in range(len(MDT)): ## To Handle the = for the parameters

split = MDT[i].split()

#print(split)

for j in range(len(split)):

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

m = ijk

#print(m)

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)): ## TO REPLACE THE PARAMETERS WITH THE THE ALA INDEX

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])))

temp=[]

for i in MNT.keys(): ##to put the calls made inside the macro in MDT

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]:

putinmdt(MDT[i],each,i)

col = 0

mdt2 = []

no\_of\_parameters = []

for i in range(len(mdt1)):

mdt2.append(mdt1[i])

#mdt1[i] = mdt1[i].replace(mdt1[i],colored(mdt1[i],colour\_list[col]))

for each in temp:

if mdt1[i].\_\_contains\_\_(each):

MNT[each]=i+col

col-=1

mdt2.pop()

no\_of\_parameters.append(len(MDT[i].split()) - 1)

#col+=1

#mdt1[i] = mdt1[i].replace(mdt1[i],colored(mdt1[i],colour\_list[col]))

MDT=mdt2

print()

##BELOW CODE FOR DISPLAY

temp = []

MNT1= {}

no\_of\_parameters\_pointer = 0

for k in MNT.keys():

temp.append(k)

MNT1[k]= [no\_of\_parameters[no\_of\_parameters\_pointer],MNT[k]+1]

no\_of\_parameters\_pointer+=1

print(" {}".format(colored("MNT" ,'red',attrs=['bold','underline','blink'])))

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

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

##TO PRINT MDT

MDTSTRUCTUREDLIST = []

i=0

for each in MDT:

MDTSTRUCTUREDLIST.insert(i,[i+1,each])

i+=1

headers = ["",colored("MDT",'red',attrs=['bold','underline','blink'])]

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

##TO PRINT Formal Parameteres v/s Positional Paramaters

print(colored(" FORMAL PARAMETERS","yellow"))

headers = ["Formal Parameteres","Positional Paramaters"]

for i in range(len(MNT)):

dicto = {}

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

dicto[function[list(MNT)[i]][j]] = "#" + str(j)

print()

res = not bool(dicto)

if "True" == str(res):

pass

else :

print(" {} {}".format("MACRO NAME : ",colored(list(MNT)[i],'red')))

print(tabulate(dicto.items(),headers= headers,tablefmt="fancy\_grid"))

##TO PRINT Actual Parameteres v/s Positional Paramaters

print(colored(" ACTUAL PARAMETERS ","yellow"))

headers = ["Actual parameters ","Positional Paramaters"]

for each in range(len(listtt)):

temp\_dict = {}

temp\_dict[listtt[each][1]]=listtt[each][2]

print(" {} {}".format("MACRO NAME : ",colored(listtt[each][0],'red')))

print(tabulate(temp\_dict.items(),headers= headers,tablefmt="fancy\_grid"))

print()

print()

print(colored("INTERMIDIATE CODE","red",attrs=['bold','underline','blink']))

file.close()

for each in left:

print(each)

**OUTPUT :-**



