LPCC Assignment 2-b

Name: Digvijay Pawar
Class: TY.Btech Comp B2
Gr.No: 21810344
Roll No: 322043

Aim: Design suitable data structures & implement pass-I for a macro with parameter.

Objective: To understand concepts of Macro with parameter.

Theory: Writing a macro is another way of ensuring modular programming in assembly language. A
macro is a sequence of instructions, assigned by a name and could be used anywhere in the
program. In NASM, macros are defined with %macro and %endmacro directives.

Code:

2B.py :

```
fhand = open('task.txt', 'r')
curr_mac = "NULL"
code = {}
para = {}
output = []

for line in fhand:
    line.strip()
```

```
    dup_line = line
    words=line.split()
    if words[0] == "MACRO":
        curr_mac = words[1]
        param = []
        for y in words[2:]:
            param.append(y)
        code[words[1]] = []
        para[words[1]] = param
    elif words[0]!="MACRO" and curr_mac=="NULL":
        output.append(dup_line)
    elif words[0] == "MEND":
        code[curr_mac].append(words)
        curr_mac = "NULL"
    elif words[0] != "MACRO" and curr_mac != "NULL":
        code[curr_mac].append(words)

mdt = []
start = {}
i = 1
actual_pram = {}

for key in code.keys():
    loop = 1
    values = {}
    for x in para[key]:
        values[x] = "#" + str(loop)
        loop = loop+1
    start[key] = i
    for x in code[key]:
        if x[0] not in code.keys():
            n = 0
            stmt = x[:]
            for element in stmt:
                if element in para[key]:
                    stmt[n] = values[element]
                n = n + 1
```

```
            temp = [i,stmt]
            mdt.append(temp)
            i = i + 1

for line in output:
    line = line.replace(","," ")
    words = line.split()
    if words[0] in para.keys():
        temp = []
        for y in words[1:]:
            temp.append(y)
        if words[0] not in actual_pram.keys():
            actual_pram[words[0]] = []
        actual_pram[words[0]].append(temp)

print("First Pass: ")
print()
print("Intemediate Code : ") #Display Intermediate Code
for x in output:
    print(x, end=" ")
print()
print("\nMacro Defination Table (MDT) : ") #Display MDT
for x in mdt:
    print(x[0],end = " ")
    for y in x[1]:
        print(y,end = " ")
    print()
print()
print("Macro Name Table(MNT) : ") #Display MNT
print("Name of Macro   | No. of para \t| Starting Index")
for x in para.keys():
    print(x,"\t\t|",len(para[x]),"\t\t\t|",start[x])

print("\nFormal vs Positional para list: \n")
for key in para.keys():
    if len(para[key]) > 0:
        print("MACRO = ",key)
```

```
        print("Formal Parameter| Positional Parameter")
        k = 1
        for x in para[key]:
            print(x,"\t\t| ","#"+str(k))
            k = k + 1
        print()
print("\nActual vs Positional para list: \n")
for key in actual_pram.keys():
    if len(para[key]) > 0:
        print("MACRO = ",key)
        for x in actual_pram[key]:
            k = 1
            print("Actual Parameter| Positional Parameter")
            for element in x:
                print(element,"\t\t| ","#"+str(k))
                k = k + 1
            print()

fhand.close()
```

task.txt:

```
START
READ A
REAB B
MACRO SUB1 S1 S2
MOVER AREG S1
SUB AREG S2
MOVEM AREG S1
MEND
MACRO ADD1 P Q
MOVER AREG P
ADD AREG Q
MOVEM AREG P
MEND
```

ADD1 9 4
SUB1 9 4
A DS 1
B DS 1
END

Output: