

Program:-

```
from sys import exit
```

```
motOpCode = ["MOV", "ADD", "SUB", "MUL", "DIV", "AND", "OR",  
             "LOAD", "STORE", "DCR", "INC", "JMP", "JNZ", "HALT"]  
keywords = ["MACRO", "CONST", "DOUBLE", "INT", "FLOAT", "SHORT", "LONG", "STRUCT", "IF",  
            "ELSE", "FOR", "SWITCH",  
            "CASE", "CHAR", "RETURN", "PRINTF", "SCANF", "AX", "BX", "CX", "DX", "AH", "BH", "CH",  
            "DH", "AL", "BL",  
            "CL", "DL"]  
sourceCode = []  
macroNames = []  
macroDefinition = []  
outputSourceCode = []  
noOfInstructionSC = 0  
noOfMacroCall = 0  
noOfInstructionMC = 0  
expandedCode = 0  
totalArgs = []  
x = 0  
mapping = {}
```

```
mc = int(input("Enter the number of Macro Definition code line : "))
```

```
for i in range(mc):
```

```
    instruction = input(  
        "Enter Macro code instruction {} :".format(i + 1)).upper()  
    macroDefinition.append(instruction)
```

```
if macroDefinition[0] == "MACRO" and macroDefinition[-1] == "MEND":
```

```
    temp = str(macroDefinition[1])  
    macroName, *argName = temp.split()  
    temp = argName  
    for i in range(len(temp)):  
        if ',' in temp[i]:  
            argName[i] = argName[i][0:-1]  
    if macroName not in keywords and macroName not in motOpCode:  
        macroNames.append(macroName)
```

```
else:
```

```
    print("Invalid Macro Definition.")  
    exit(0)
```

```
sc = int(input("Enter the number of Source code lines : "))
```

```
for i in range(sc):
```

```
    instruction = input(  
        "Enter Source code instruction {} : ".format(i + 1)).upper()  
    sourceCode.append(instruction)
```

```
for i in range(sc):
```

```
    if macroName in sourceCode[i]:
```

```

        noOfMacroCall = noOfMacroCall + 1
    else:
        noOfInstructionSC = noOfInstructionSC + 1

for i in range(sc):
    if macroName in sourceCode[i]:
        x = x + 1
        noOfInstructionMC = 0
        temp = str(sourceCode[i])
        macroName, *argValue = temp.split()
        totalArgs.append(argValue)
        temp = argValue
        for j in range(len(temp)):
            if ',' in temp[j]:
                argValue[j] = argValue[j][0:-1]

        # Create Dictionary for mapping
        for j in range(len(argName)):
            name, value = argName[j], argValue[j]
            mapping[name + str(x)] = value

        for j in range(2, mc - 1):
            for k in range(len(argName)):
                if argName[k] in macroDefinition[j]:
                    temp = macroDefinition[j]
                    opCode, value = temp.split()
                    tempValue = mapping.get(value + str(x))
                    temp = opCode + ' ' + str(tempValue)
                    outputSourceCode.append(temp)
                    noOfInstructionMC = noOfInstructionMC + 1
            else:
                temp = sourceCode[i]
                outputSourceCode.append(temp)

print("Expanded Source Code is : ")
for i in outputSourceCode:
    print(i)
    expandedCode = expandedCode + 1

print()
print("No of instructions in input source code : {}".format(noOfInstructionSC))
print("No of macro call : {}".format(noOfMacroCall))
print("No of instructions defined in macro call : {}".format(noOfInstructionMC))
for i in range(len(totalArgs)):
    print("Actual argument during {} Macro call 'JUHILEE' = {}".format(i + 1, ', '.join(totalArgs[i])))
    print("Total number of instructions in expanded code : {}".format(expandedCode))

```

Output:

```
Enter the number of Macro Definition code line : 6
Enter Macro code instruction 1 :MACRO
Enter Macro code instruction 2 :JUHILEE &ARG1, &ARG2, &ARG3
Enter Macro code instruction 3 :ADD &ARG1
Enter Macro code instruction 4 :SUB &ARG2
Enter Macro code instruction 5 :OR &ARG3
Enter Macro code instruction 6 :MEND
Enter the number of Source code lines : 7
Enter Source code instruction 1 : MOV R
Enter Source code instruction 2 : JUHILEE 20, 30, 40
Enter Source code instruction 3 : DCR R
Enter Source code instruction 4 : AND R
Enter Source code instruction 5 : JUHILEE 22, 33, 44
Enter Source code instruction 6 : MUL 88
Enter Source code instruction 7 : HALT
```

Expanded Source Code is :

```
MOV R
ADD 20
SUB 30
OR 40
DCR R
AND R
ADD 22
SUB 33
OR 44
MUL 88
HALT
```

```
No of instructions in input source code : 5
No of macro call : 2
No of instructions defined in macro call : 3
Actual argument during 1 Macro call 'JUHILEE' = 20, 30, 40
Total number of instructions in expanded code : 11
Actual argument during 2 Macro call 'JUHILEE' = 22, 33, 44
Total number of instructions in expanded code : 11
```