MACRO PROCESSOR

```
import re
f_input = open("macro_input.txt")
inputcode = list(line.strip() for line in f_input)


MDT = []
MNT = {}
ALA_list = {}  # ALA_list stores ALA for each macro, with the macro name as key
input_for_pass_2 = [
] iterator = iter(inputcode)
while True:
    try:
        line = next(iterator) if line
        == "MACRO":
            nameline = next(iterator) nameline =
            re.split('[,\s]', nameline) macro_name = "" for
            token in nameline:
                if "&" not in token:
                    macro_name = token break

            MNT[macro_name] =
            len(MDT) ALA = {}
            arg_counter = 0 for token in
            nameline:
                if token is not macro_name:
                    arg_counter += 1
                    ALA[token] = "#" + str(arg_counter)
                    nameline[nameline.index(token)] = ALA[ token]
            ALA_list[macro_name] = ALA
            MDT.append(nameline)

            while True:
                macroline = next(iterator) for
                argument in ALA.keys():
                    if argument in macroline:
                                macroline = macroline.replace(argument, ALA[argument])
                MDT.append(macroline) if
                macroline == "MEND":
                    break else:
```

```
                input_for_pass_2.append(line)
        except StopIteration:
break print("\nMNT is ")

for line in MNT.items():
        print(line)
print("\nMDT is ") for
line in MDT: print(line)
print("\nALAs are ") for line in
ALA_list.items(): print(line)


iterator = iter(input_for_pass_2)
print("\n Final Output is ") while True:
try:
            line = next(iterator) line = re.split('[,\s]', line) if
            any(word in line for word in MNT.keys()):
            macroname = "" if line[0] in MNT.keys():
                    macroname = line[0]
                else:
                    macroname = line[1] label =
                    line[0]
                actual_args = [] for token in line: if
                    not token == macroname:
                            actual_args.append(token)
                                ALA = ALA_list[macroname]
                ALA = {val: key for key, val in
                        ALA.items()
                        } formal_args =
                sorted(list(ALA.keys())) for i in
                range(len(formal_args)):
                    ALA[formal_args[i]] = actual_args[ i]
                MDTP = MNT[macroname] + 1
                while "MEND" not in
                MDT[MDTP]:
                    line = MDT[MDTP] for formal_arg, actual_arg in
                    ALA.items(): line = line.replace(formal_arg,
                    actual_arg)
                    print(line)
                    MDTP += 1
            else:
```

```
                print(" ".join(line))
        except StopIteration:
            break
```

Sample Code :

```
        MACRO
        INCR &ARG1
        L AX,&ARG1
        A AX,1
        MEND
        MACRO
        FOOBAR &ARG1,&ARG2
        L AX,&ARG1
        L BX,&ARG2
        ST AX,BX
        MEND
        MACRO
        &LAB HARAMBE &ARG1
        &LAB SR &ARG1,1
        RR &ARG1,2
        MEND
        START 0
        INCR 69
        FOOBAR 69,96
        LOOP HARAMBE 69
        DC F'69'
        END
```

Output:

```
MNT is
('INCR', 0)
('FOOBAR', 4)
('HARAMBE', 9)

MDT is
['INCR', '#1']
L AX,#1
A AX,1
MEND
['FOOBAR', '#1', '#2']
L AX,#1
L BX,#2
ST AX,BX
MEND
['#1', 'HARAMBE', '#2']
#1 SR #2,1
RR #2,2
MEND

ALAs are
('INCR', {'&ARG1': '#1'})
('FOOBAR', {'&ARG1': '#1', '&ARG2': '#2'})
('HARAMBE', {'&LAB': '#1', '&ARG1': '#2'})
```

```
 Final Output is
START 0
L AX,69
A AX,1
L AX,69
L BX,96
ST AX,BX
LOOP SR 69,1
RR 69,2
DC F'69'
END
```