**Mnt , mnd**# Open input and output files

code = open("input.txt", "r") # Open the input file for reading (contains assembly code with macros)

mnt = open("mnt.txt", "a+") # Open the MNT (Macro Name Table) file for appending

mdt = open("mdt.txt", "a+") # Open the MDT (Macro Definition Table) file for appending

fpvpp = open("fpvpp", "a+") # Open the FPVPP (Formal Parameters and Actual Parameters) file for appending

mnt.write("Macro\_Name \t Start-End \n") # Write header for the MNT table

# Initialize lists to store MNT and MDT information

mntable = [] # List to store the macro names and their details in MNT

mdtable = [] # List to store the macro definitions in MDT

# Counters for tracking MNT and MDT entries

mntp = 0 # Counter for MNT (Macro Name Table)

mdtp = 0 # Counter for MDT (Macro Definition Table)

# Flag to indicate whether we are inside a macro definition or not

flag = False # Initially, not inside a macro definition

# Read all lines from the input code

lines = code.readlines()

# Process each line from the input code

for l in lines:

x = l.split() # Split the line into tokens (based on spaces)

if len(x) > 0: # Check if the line contains any tokens

# Check if the line defines a new macro

if x[0] == "MACRO":

flag = True # Set the flag to True, indicating we are inside a macro definition

mntp += 1 # Increment the MNT counter

mntable.append(x[1]) # Add the macro name to the MNT table

mntable.append(mdtp) # Add the starting MDT index to the MNT table

if len(x) > 2: # If there are parameters for the macro

# Compute the index for the formal parameters in FPVPP

l = mntp \* 3

fpvpp.write(mntable[l - 3] + "\n") # Write the macro name to the FPVPP file

# Write the formal parameters with their indices (starting from 1)

for i in range(2, len(x)):

fpvpp.write(x[i] + " " + str(i - 1) + "\n") # Write each parameter and its index

fpvpp.write("\n") # Add a newline after the parameters

# Check if the line ends a macro definition (MEND)

elif x[0] == "MEND":

flag = False # Set the flag to False, indicating the end of the macro definition

mntable.append(mdtp - 1) # Add the ending MDT index to the MNT table

# Write the macro's details (name, start MDT index, and end MDT index) to the MNT file

l = mntp \* 3

mnt.write(mntable[l - 3] + " " + str(mntable[l - 2]) + " " + str(mntable[l - 1]) + "\n")

# If we are inside a macro definition (flag == True)

elif flag == True:

# Write each part of the macro definition to the MDT file

for i in x:

mdt.write(str(i) + " ") # Write each token in the macro definition

mdt.write(" " + str(mdtp) + "\n") # Write the MDT index

mdtp += 1 # Increment the MDT counter for the next line