Algorithm for one - pass macro processor

```
begin {macro processor}
    EXPANDING := FALSE
   while OPCODE != 'END' do
      begin
         GETLINE
         PROCESSLINE
       end {while}
   end{macro processor}


procedure GETLINE
   begin
     if EXPANDING then
       begin
         get next line of macro definition from DEFTAB
        substitute argument from ARGTAB for positional notation
        end {if}
       else
        read next line from input file
 end {GETLINE}



procedure PROCESLINE
     begin
       search NAMTAB for OPCODE
       if found then
           EXPAND
       else if OPCODE = 'MACRO' then
           DEFINE
     else write source line to expanded file
end   {PROCESSLINE}
```

```
procedure EXPAND
        begin
          EXPANDING := TRUE
          get first line of macro definition {prototype}from DEFTAB
          set up argument from macro invocation in ARGTAB
           write macro invocation to expanded file as a comment
           while not end of macro definition do
               begin
                  GETLINE
                  PROCESSLINE
               end {while}
             EXPANDING := FALSE
         end {EXPAND}



procedure DEFINE
    begin
        enter macro name into NAMTAB
        enter macro prototype into DEFTAB
        LEVEL := 1
       while LEVEL > 0 do
          begin
            GETLINE
           if this is not a comment line then
               begin
                  substitute positional notation for parameters
              enter line into DEFTAB
              if OPCODE = 'MACRO' then
                        LEVEL := LEVEL +1
               else if OPCODE = 'MEND' then
                      LEVEL := LEVEL -1
         end {if not comment}
       end {while}
  store in NAMTAB pointer to beginning and end of definition
end {DEFINE}
```