* + - **Single Pass Assembler Algorithm**

Begin

Read 1st input line

If *OPCODE = ’START’* then

{

Starting address = #OPERAND

LOCCTR = Starting address

Read the next input line

}

Else

LOCCTR = 0

Create Header Record and write it to object program

Initialize 1st Text Record

While *OPCODE != ‘END’* do

{

If *there is not a comment line* then

{

If *there is a symbol in the LABEL field* then

{

Search SYMTAB for LABEL

If *found* then

{

If *symbol value as null* then

{

Symbol value = LOCCTR

Search the linked list with corresponding operand

Generate operand addresses as corresponding to symbol value

Delete the linked list

}

}

Else

Insert (LABEL, LOCCTR) into SYMTAB

}

Search OPTAB for OPCODE

If *found* then

{

Search SYMTAB for OPERAND address

If *found* then

{

If *symbol value != null* then

OPERAND address = symbol value

Else

Insert a node at the end of the linked list with address as LOCCTR

}

Else

Insert (symbol name, null) into SYMTAB

Generate object code

LOCCTR = LOCCTR +3

}

Else if *OPCODE = ‘WORD’* then

{

LOCCTR = LOCCTR +3

Object code = #OPERAND

}

Else if *OPCODE = ‘RESW’* then

LOCCTR = LOCCTR +3

Else if *OPCODE = ‘RESB’* then

LOCCTR = LOCCTR + #OPERAND

Else if *OPCODE = ‘BYTES’* then

{

LOCCTR = LOCCTR +length of the constant

Convert constant to object code

}

Else

Set error flag

If *object code will not fit into the current text record* then

{

Write Text Record into object program

Initialize new Text Record

}

Add object code to Text Record

}

Write listing line

Read the next input line

}

Write last listing line

Write last Text Record to object program

Write End Record to object program

End