Requirements:

* design a two pass assembler for a specific machine
* Include test procedures and data.
* How to use the program.
* describe any technical details that may need maintenance

Introduction:

First of all an assembler is basically a translator that translates an assembly language code into machine language. Two pass assembler does the same work with a little bit variation of mechanism from that of one pass assembler.

Two pass assembler:

A two pass assembler does two passes over the source file (the second pass can be over a file generated in the first pass). In the first pass all it does is looks for label definitions and introduces them in the symbol table. In the second pass, after the symbol table is complete, it does the actual assembly by translating the operations and so on.

First pass:

* Assign addresses to all the statement
* Addresses of symbolic labels are stored
* Some assemble directives will be processed

Second pass:

* Translate opcodes and symbolic operands
* Generate data values defined by byte and words
* Write the object program and assembly listing

Difference between one pass and two pass assembler:

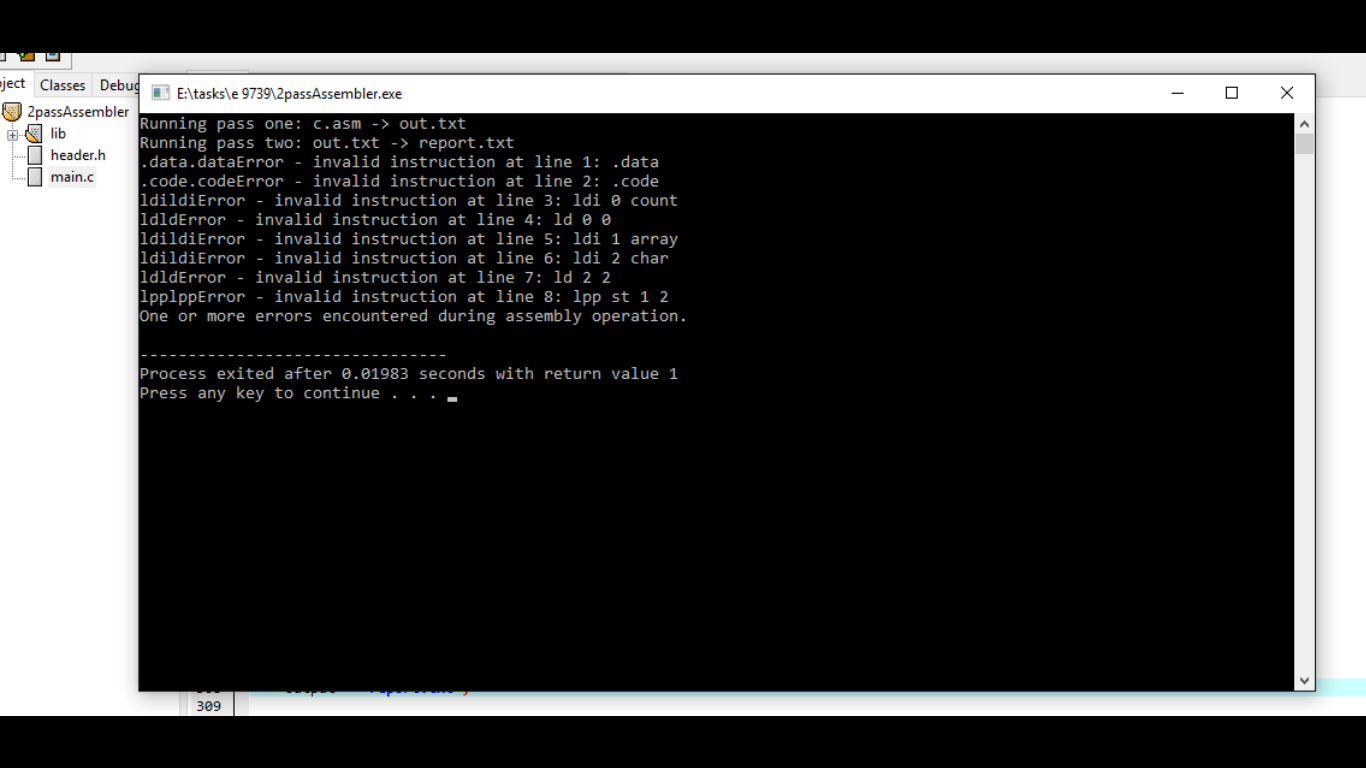
The difference between one pass and two pass assemblers are:  
 A one pass assembler passes over the source file exactly once, in the same pass collecting the labels, resolving future references and doing the actual assembly. The difficult part is to resolve future label references and assemble code in one pass.  
 A two pass assembler does two passes over the source file (the second pass can be over a file generated in the first pass). In the first pass all it does is looks for label definitions and introduces them in the symbol table. In the second pass, after the symbol table is complete, it does the actual assembly by translating the operations and so on.

Objectives:

The main objective of this project is to become familiar with the concept that how to design a two pass assembler for a specific machine.

Experiment:

Output:



.data  
.code  
ldi 0 count  
ld 0 0  
ldi 1 array  
ldi 2 char  
ld 2 2  
lpp st 1 2

.text  
  
.symbol  
4 count  
4 array  
4 char  
  
.relocation

User manual:

Write a code in c.asm file that is comment free and space free then run the assembler. This will automatically create the log, out and report file that are the summary and also display some of results of processing. The above experiment result is the output generated in files that are used for report purposes

The c.asm file code must be comment free and have no blank line at end or between must be a basic assembly code