

# USER MANUAL

BY Antony Jacob

# A Comprehensive Guide to the Two-Pass Assembler

---

## Introduction

Welcome to the Two-Pass Assembler Web Tool, designed to streamline the process of converting assembly language into machine code through the two-pass assembly process. This tool allows you to upload your assembly source file and opcode table (optab), process them through Pass 1 and Pass 2 algorithms, and generate outputs such as intermediate code, symbol tables, program length, and object code.

Whether you're a student or a professional, this tool offers a clear and intuitive interface for managing assembly code compilation.

---

## Features of the Website

### 1. File Uploads:

- Allows uploading of two critical files: the assembly input source file and the opcode table (optab) file.

### 2. Pass One Processing:

- Executes the pass one of the two-pass assembler to generate the intermediate output, SYMTAB (symbol table), and program length.

### 3. Pass Two Processing:

- Executes the pass two, producing the final overall output and object code, which is the machine code for the input assembly program.

### 4. Downloadable Outputs:

- Users can download generated outputs like intermediate files, symbol tables, program lengths, and object codes as .txt files.

### 5. Clear Segregation of Pass One and Pass Two:

- The site is structured to distinctly handle Pass 1 and Pass 2 operations through individual tabs for clarity.

## 7. The Two-Pass Assembler Algorithms:

- There is a separate section that displays algorithm for each pass one and pass two of two pass assembler.
- 

## How to Use This Website

### Step-by-Step Guide:

#### 1. Visit the Home Page:

- Upon loading the website, you'll be greeted with a simple, user-friendly interface. The navbar allows you to navigate between the **Home**, **Pass One**, and **Pass Two** sections.

#### 2. Uploading Files:

- In the "Upload Files" section, two file inputs are required:
- **Input File:** Upload a .txt file containing the assembly code you want to process.
- **Optab File:** Upload a .txt file containing the opcode table (optab), which maps operation codes to machine instructions.

#### 3. Running the Assembler:

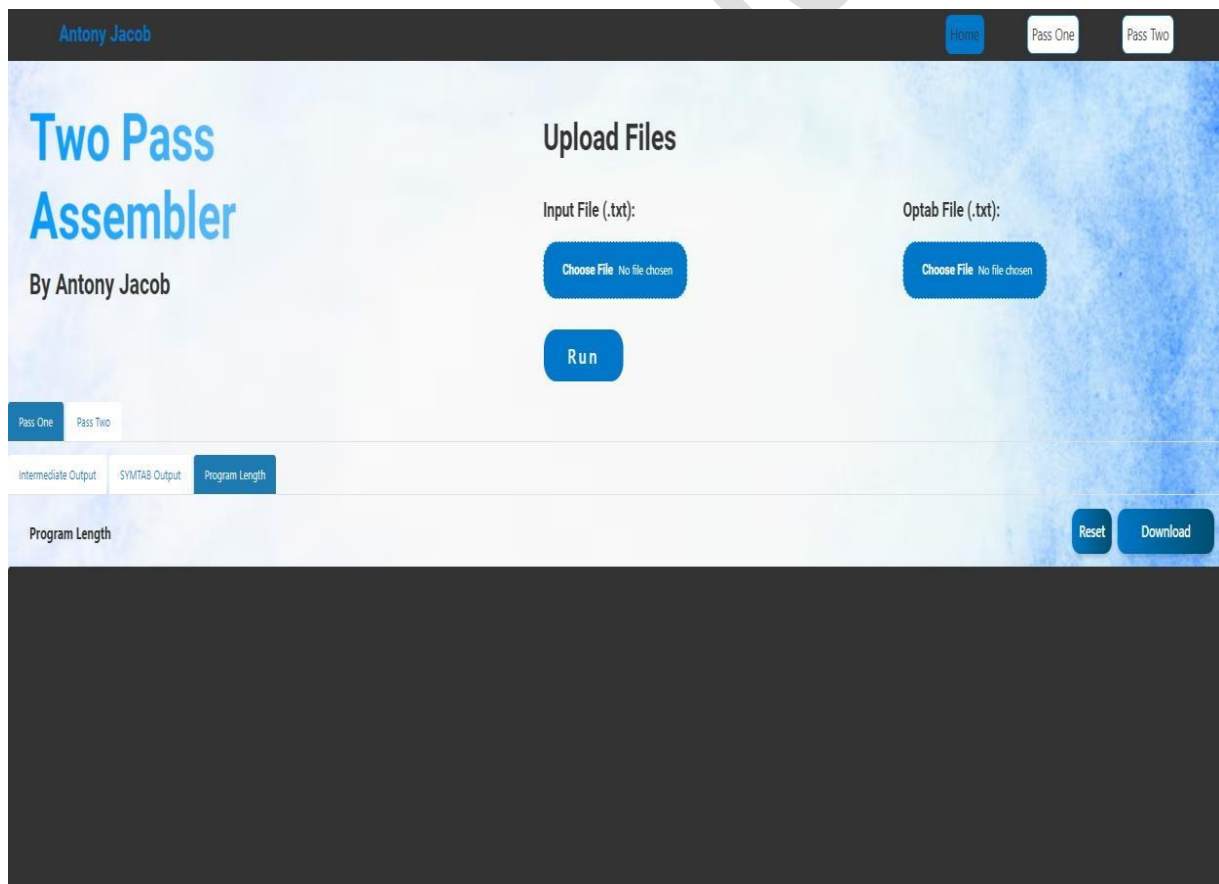
- Once both files are uploaded, click the **Run** button to initiate the two-pass assembler process.

#### 4. Viewing Pass One Outputs:

- Navigate to the **Pass One** tab to view the intermediate outputs. The three sections within Pass One are:
- **Intermediate Output:** Displays the intermediate file generated during Pass 1.
- **SYMTAB Output:** Shows the symbol table with the addresses of the symbols used.

- **Program Length:** Displays the total length of the program after Pass 1.
5. **Viewing Pass Two Outputs:**
- Go to the **Pass Two** tab to check the final outputs. You'll find:
  - **Output:** The overall machine code generated after Pass 2.
  - **Object Code:** The final object code for each instruction in header record, text record and end record (HTE).
6. **Downloading Results:**
- You can download the results in .txt format for further analysis or use by clicking the **Download** buttons next to each output section.
- 

## User Interface



The screenshot displays the web interface of the 'Two Pass Assembler' by Antony Jacob. The top navigation bar includes the user's name 'Antony Jacob' and three tabs: 'Home' (active), 'Pass One', and 'Pass Two'. The main content area features the title 'Two Pass Assembler' and the author 'By Antony Jacob'. Under the 'Upload Files' section, there are two file upload areas: 'Input File (.txt):' and 'Optab File (.txt):', each with a 'Choose File' button and a 'No file chosen' status. A 'Run' button is positioned below these. Below the upload section, there are two tabs: 'Pass One' and 'Pass Two'. Under the 'Pass Two' tab, there are three sub-tabs: 'Intermediate Output', 'SYMTAB Output', and 'Program Length' (which is currently selected). The 'Program Length' sub-tab shows the text 'Program Length' and two buttons: 'Reset' and 'Download'. The bottom portion of the interface is a large, dark grey rectangular area, likely intended for displaying the output of the assembly process.

## Example Input and Optab

Sample Input File (input.txt):

PGM1	START	1000
-	LDA	ALPHA
-	MUL	BETA
-	STA	GAMMA
ALPHA	WORD	2
BETA	WORD	4
GAMMA	RESW	1
-	END	1000

Sample Optab File (optab.txt):

LDA	00
STA	0C
MUL	20

---

# Home Page

## Intermediate Output:

This file contains information about each line of code and its associated address, generated after pass 1.

The screenshot shows the 'Two Pass Assembler' web application. The 'Home' tab is selected. Under 'Upload Files', there are buttons for 'Choose File' for 'input.txt' and 'optab.txt', and a 'Run' button. Below the upload section, there are tabs for 'Pass One' and 'Pass Two'. Under 'Pass One', there are tabs for 'Intermediate Output', 'SYMTAB Output', and 'Program Length'. The 'Intermediate Output' tab is active, displaying a table of assembly code with addresses, labels, instructions, and values. 'Reset' and 'Download' buttons are visible on the right.

	PGM1	START	1000
1000	-	LDA	ALPHA
1003	-	MUL	BETA
1006	-	STA	GAMMA
1009	ALPHA	WORD	2
100c	BETA	WORD	4
100f	GAMMA	RESW	1
1012	-	END	1000

## SYMTAB Output:

This output shows the symbol table (SYMTAB) which includes labels used in the input file and their assigned addresses.

The screenshot shows the same web application, but the 'SYMTAB Output' tab is active under the 'Pass One' section. It displays a symbol table with labels and their assigned addresses. 'Reset' and 'Download' buttons are visible on the right.

ALPHA	1009
BETA	100c
GAMMA	100f

## Program Length:

Displays the total length of the program.

The screenshot shows the 'Two Pass Assembler' web application. The 'Home' button is selected in the top navigation bar. The 'Upload Files' section has 'Choose File' buttons for 'input.txt' and 'optab.txt', and a 'Run' button. Below this, the 'Pass One' and 'Pass Two' tabs are visible, with 'Pass Two' being the active tab. The 'Program Length' tab is selected in the output section. The output displays 'The length of the program is: 0012'. There are 'Reset' and 'Download' buttons in the top right of the output area.

Antony Jacob

Home Pass One Pass Two

# Two Pass Assembler

By Antony Jacob

Upload Files

Input File (.txt):

Choose File input.txt

Optab File (.txt):

Choose File optab.txt

Run

Pass One Pass Two

Intermediate Output SYMTAB Output Program Length

Program Length

Reset Download

The length of the program is: 0012

## Output:

The overall machine code generated after Pass 2.

The screenshot shows the 'Two Pass Assembler' web application. The 'Home' button is selected in the top navigation bar. The 'Upload Files' section has 'Choose File' buttons for 'input.txt' and 'optab.txt', and a 'Run' button. Below this, the 'Pass One' and 'Pass Two' tabs are visible, with 'Pass Two' being the active tab. The 'Object Code' tab is selected in the output section. The output displays a table of machine code. There are 'Reset' and 'Download' buttons in the top right of the output area.

Antony Jacob

Home Pass One Pass Two

# Two Pass Assembler

By Antony Jacob

Upload Files

Input File (.txt):

Choose File input.txt

Optab File (.txt):

Choose File optab.txt

Run

Pass One Pass Two

Output Object Code

Output

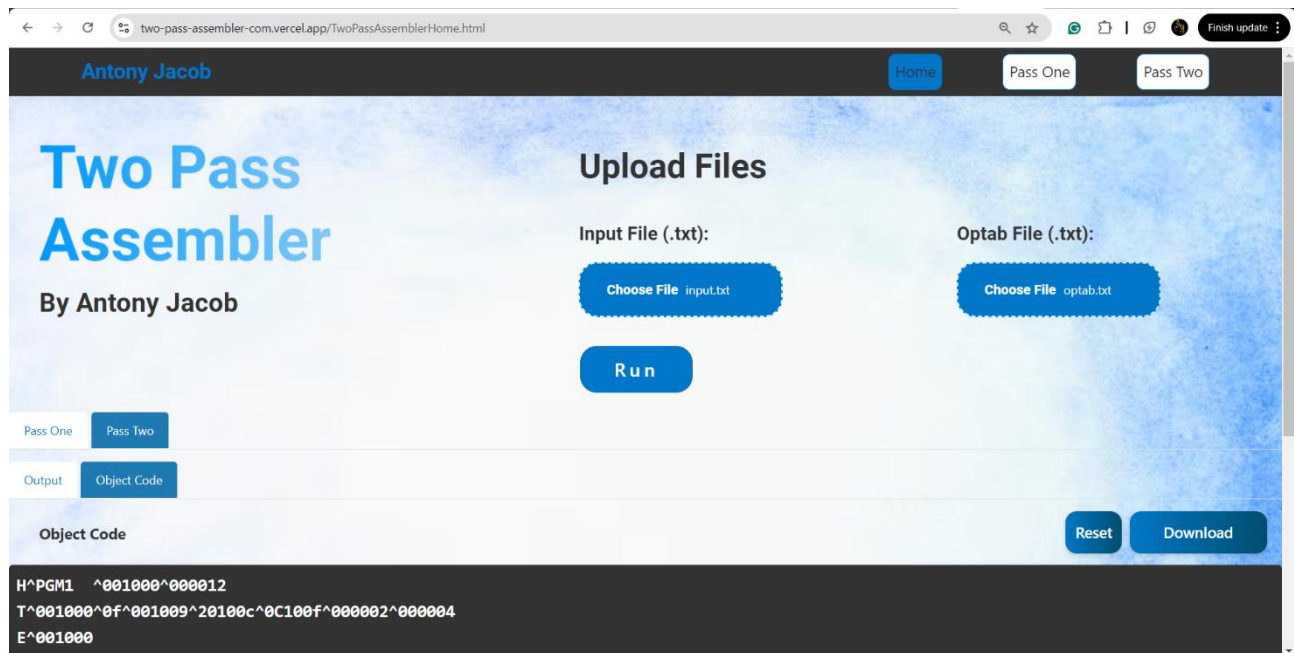
Reset Download

	PGM1	START	1000	
1000	-	LDA	ALPHA	001009
1003	-	MUL	BETA	20100c
1006	-	STA	GAMMA	0C100f
1009	ALPHA	WORD	2	000002
100c	BETA	WORD	4	000004
100f	GAMMA	RESW	1	
1012	-	END	1000	



## Object Code:

The final object code for each instruction, generated after Pass 2.

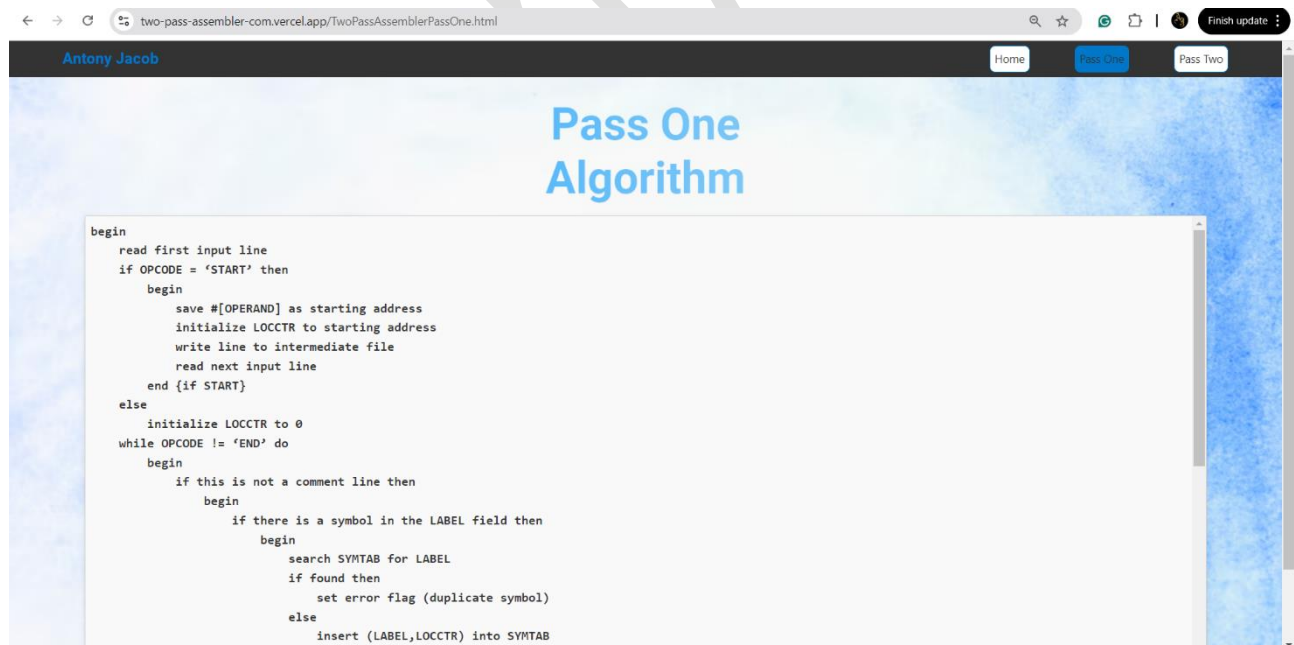


The screenshot shows the 'Two Pass Assembler' home page by Antony Jacob. It features a navigation bar with 'Home', 'Pass One', and 'Pass Two' buttons. The main heading is 'Two Pass Assembler' with the subtitle 'By Antony Jacob'. Below this, there are two sections for file uploads: 'Input File (.txt)' and 'Optab File (.txt)', each with a 'Choose File' button. A 'Run' button is positioned below the input file section. At the bottom, there are tabs for 'Output' and 'Object Code', with 'Object Code' currently selected. The 'Object Code' section displays the following hex code:   
H^PGM1 ^001000^000012  
T^001000^0f^001009^20100c^0C100f^000002^000004  
E^001000  
Buttons for 'Reset' and 'Download' are located to the right of the object code display.

## Pass One Page

### Pass One Algorithm:

This page contains the algorithm for Pass One of the Two-Pass Assembler.



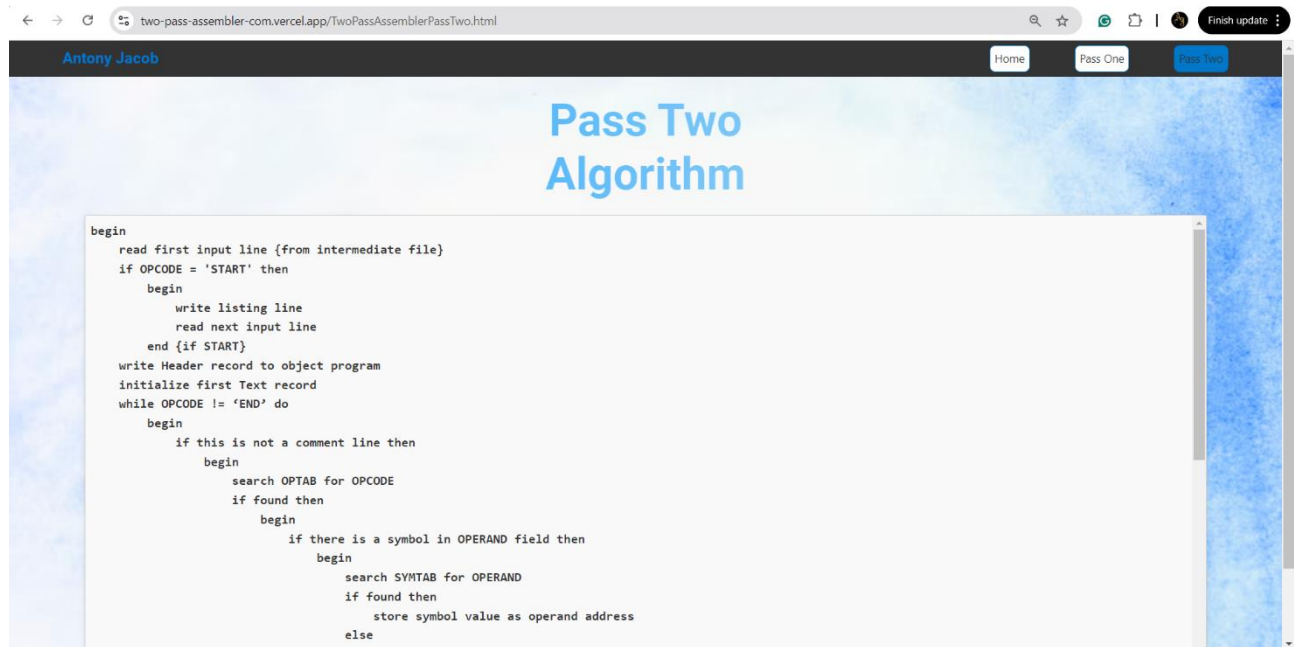
The screenshot shows the 'Pass One Algorithm' page. It features a navigation bar with 'Home', 'Pass One', and 'Pass Two' buttons. The main heading is 'Pass One Algorithm'. Below the heading, there is a code editor displaying the following algorithm:   
begin  
  read first input line  
  if OP CODE = 'START' then  
    begin  
      save #[OPERAND] as starting address  
      initialize LOCCTR to starting address  
      write line to intermediate file  
      read next input line  
    end (if START)  
  else  
    initialize LOCCTR to 0  
  while OP CODE != 'END' do  
    begin  
      if this is not a comment line then  
        begin  
          if there is a symbol in the LABEL field then  
            begin  
              search SYMTAB for LABEL  
              if found then  
                set error flag (duplicate symbol)  
              else  
                insert (LABEL,LOCCTR) into SYMTAB  
            end  
          end  
        end  
      end  
    end  
  end  
end



# Pass Two Page

## Pass Two Algorithm:

This page contains the algorithm for Pass Two of the Two-Pass Assembler.

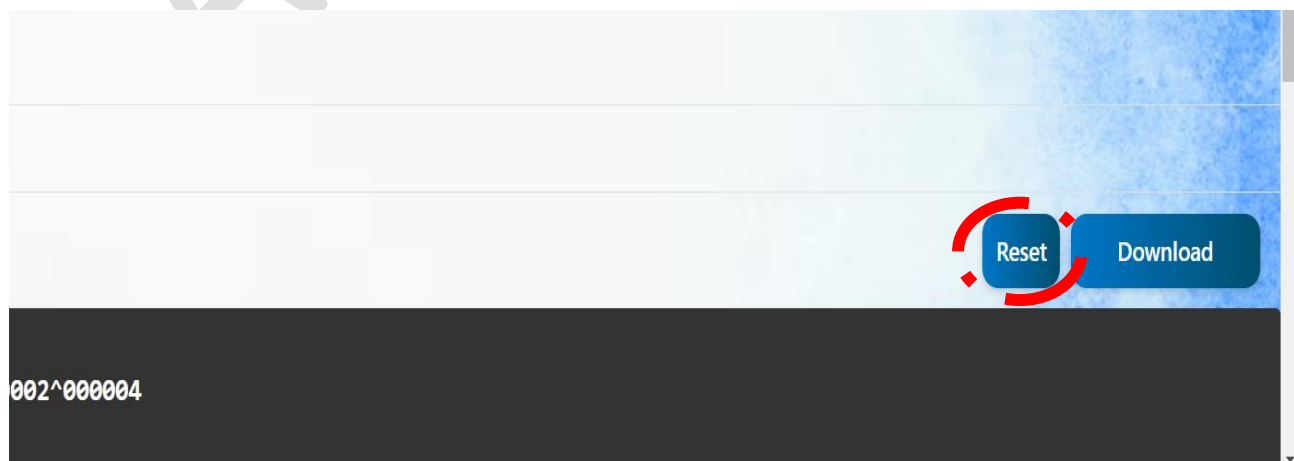


```
begin
  read first input line {from intermediate file}
  if OPCODE = 'START' then
    begin
      write listing line
      read next input line
    end {if START}
  write Header record to object program
  initialize first Text record
  while OPCODE != 'END' do
    begin
      if this is not a comment line then
        begin
          search OPTAB for OPCODE
          if found then
            begin
              if there is a symbol in OPERAND field then
                begin
                  search SYMTAB for OPERAND
                  if found then
                    store symbol value as operand address
                  else
```

## Add Ons

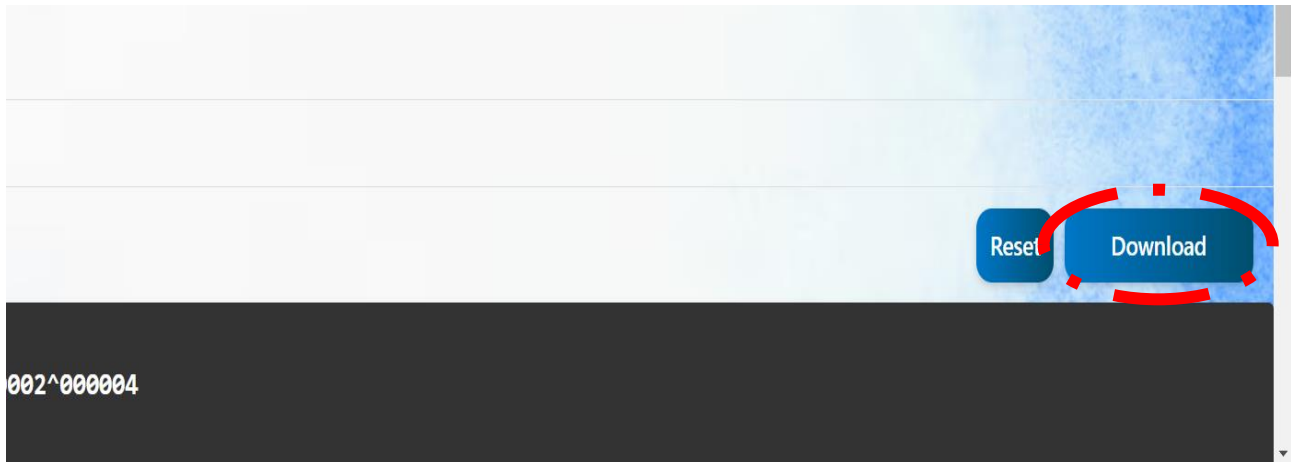
### Extra Features

- Reset or Restart: After finishing a process, you can restart the workflow by refreshing the page or clicking the restart option.
- Compatibility: Works on modern browsers (Chrome, Firefox, Edge).



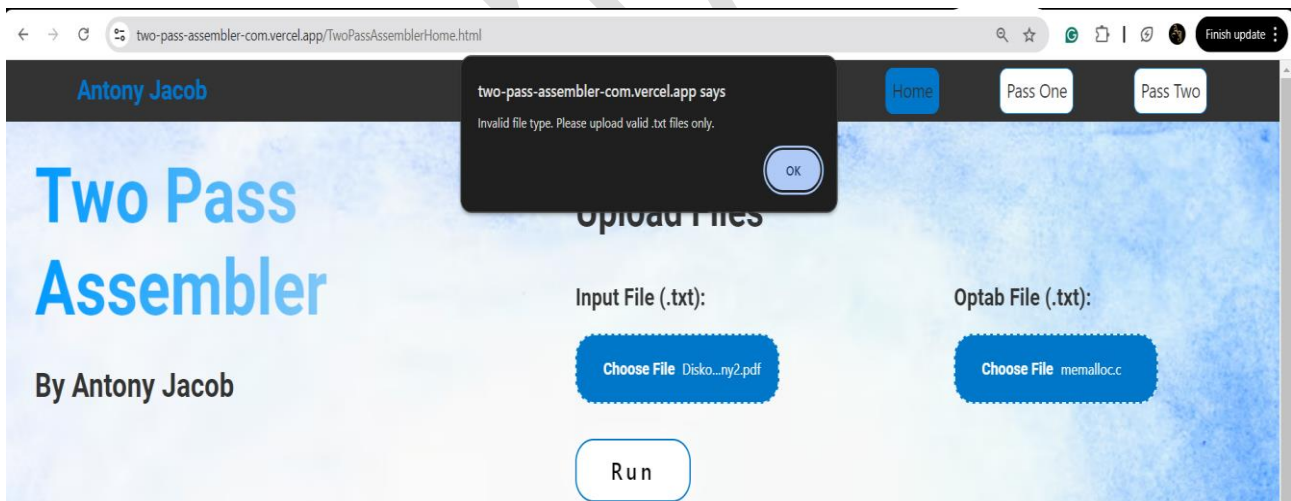
## Tips for Effective Use

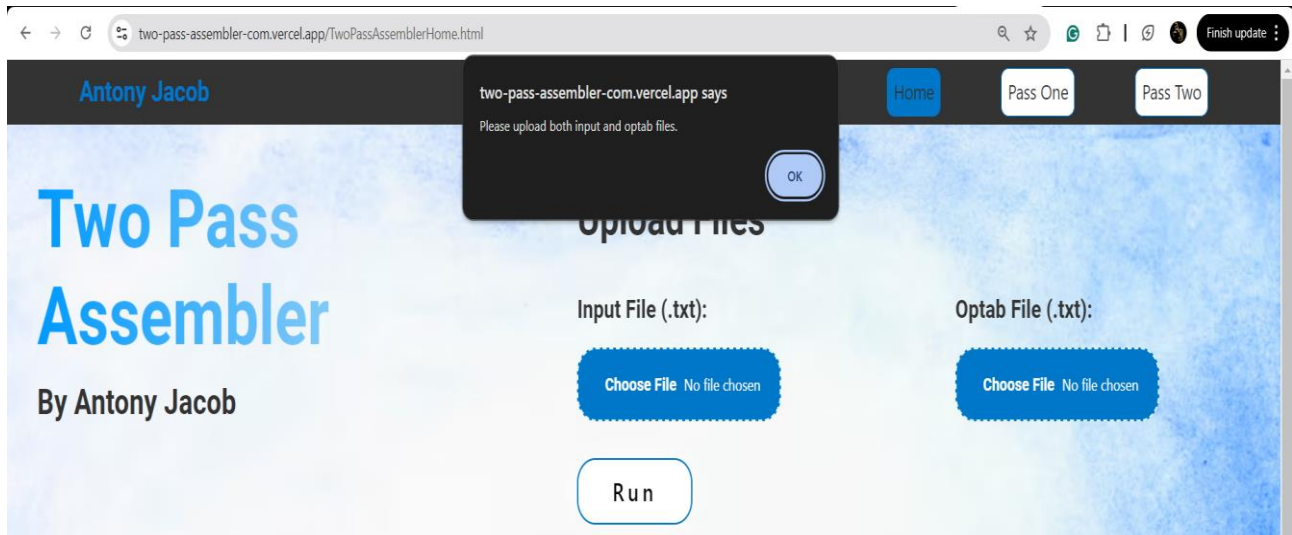
- Ensure correct formatting of the input files.
- Review the intermediate code thoroughly before moving to Pass 2 to avoid errors.
- Use the download feature to keep a local copy of all outputs.



## Troubleshooting

- File Upload Issues: Ensure that the files are correctly formatted and of the proper type. If no file is uploaded, the process will not proceed.





- No Output in Pass 1: Check if the input assembly file is correctly structured.
- Object Code Errors: If Pass 2 generates incorrect results, revisit Pass 1 and ensure the intermediate code and symbol table were properly generated.

## Conclusion

The Two-Pass Assembler web tool simplifies the task of assembling programs by providing an intuitive interface for performing both Pass 1 and Pass 2 operations. It generates crucial outputs such as intermediate files, symbol tables, and object codes, all available for download. Whether you are learning assembly language or working with real-world projects, this tool serves as a vital asset in understanding and generating machine code from assembly instructions.

Take advantage of the structured Pass One and Pass Two processes to turn your assembly language programs into executable object code, all from the comfort of your web browser!