**Testing Plan**

Contents

[Introduction 1](#_Toc285992288)

[Error Checking 1](#_Toc285992289)

[1. Test Explanation 1](#_Toc285992290)

[2. Test Results 1](#_Toc285992291)

[a. File IO 1](#_Toc285992292)

[b. Next error 1](#_Toc285992293)

[Object File Assembly 2](#_Toc285992294)

[1. Test Explanation 2](#_Toc285992295)

[2. Test Results 2](#_Toc285992296)

[a. Prompt Given Example 2](#_Toc285992297)

[b. Absolute Program 2](#_Toc285992298)

[c. Relative Program 2](#_Toc285992299)

[d. Absolute Program Using Each Machine and Pseudo-op 3](#_Toc285992300)

[e. Relative Program Using Each Machine and Pseudo-op 3](#_Toc285992301)

[f. Program with Maximum Symbols, Literals, and Source Records 3](#_Toc285992302)

[g. Program Using Forward Referencing 3](#_Toc285992303)

[h. Program Beginning Execution at Last Memory Address 3](#_Toc285992304)

[i. Program Spanning Page Boundary 3](#_Toc285992305)

# Introduction

This document describes the manner, expected output, and result of each test conducted on the assembler. The assembler will be tested to see that the expected errors are produced and also tested that the assembler produces the correct and expected object file output.

# Error Checking

## Test Explanation

The tests following are specifically designed to make the assembler produce a certain error. The error and the conditions required to produce each error are listed in the following section.

## Test Results

### File IO

Call: >java Assembler.Main doesnotexist.asm

Result: Failed to assemble program due to an IO error.

### Next error

# Object File Assembly

## Test Explanation

The tests following are assembly language files written to produce correct and expected object file output. The assembly files will be written with various lengths and complexity to test the various commands allowed in our assembly language architecture.

## Test Results

### Prompt Given Example

|  |  |
| --- | --- |
| Assembly | Object File |
| ; Example Program  Lab2EG .ORIG x30B0  count .FILL #4  Begin LD ACC,count ;R1 <- 4  LEA R0,msg  loop TRAP x22 ;print "hi! "  ADD ACC,ACC,#-1 ;R1--  BRP loop  JMP Next  msg .STRZ "hi! "  Next AND R0,R0,x0 ;R0 <- 0  NOT R0,R0 ;R0 <- xFFFF  ST R0,Array ;M[Array] <-xFFFF  LEA R5,Array  LD R6,=#100 ;R6 <= #100  STR R0,R5,#1  TRAP x25  ACC .EQU #1  ; ----- Scratch Space -----  Array .BLKW #3  .FILL x10  .END Begin | HLab2EG30B00018  T30B00004  T30B122B0  T30B2E0B7  T30B3F022  T30B4127F  T30B502B3  T30B640BC  T30B70068  T30B80069  T30B90021  T30BA0020  T30BB0000  T30BC5020  T30BD903F  T30BE30C3  T30BFEAC3  T30C02CC7  T30C17141  T30C2F025  T30C60010  T30C70064  E30B1 |
| **Result:** | Correct except for segment length |

### Absolute Program

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Relative Program

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Absolute Program Using Each Machine and Pseudo-op

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Relative Program Using Each Machine and Pseudo-op

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Program with Maximum Symbols, Literals, and Source Records

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Program Using Forward Referencing

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Program Beginning Execution at Last Memory Address

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |

### Program Spanning Page Boundary

|  |  |
| --- | --- |
| Assembly | Object File |
| .ORIG x3000  .END Begin | H30000003  T30000004  E3001 |
| **Result:** |  |