**Testing Plan**

Contents

[Introduction 1](#_Toc286066958)

[Error Checking 1](#_Toc286066959)

[1. Test Explanation 1](#_Toc286066960)

[2. Test Results 1](#_Toc286066961)

[a. File IO 1](#_Toc286066962)

[b. Next error 1](#_Toc286066963)

[Object File Assembly 2](#_Toc286066964)

[1. Test Explanation 2](#_Toc286066965)

[2. Test Results 2](#_Toc286066966)

[a. Prompt Given Example – Absolute Program 2](#_Toc286066967)

[b. Relative Program 2](#_Toc286066968)

[c. Absolute Program Using Each Machine and Pseudo-op 2](#_Toc286066969)

[d. Relative Program Using Each Machine and Pseudo-op 2](#_Toc286066970)

[e. Program with Maximum Symbols, Literals, and Source Records 3](#_Toc286066971)

[f. Program Using Forward Referencing 3](#_Toc286066972)

[g. Program Beginning Execution at Last Memory Address 3](#_Toc286066973)

[h. Program Spanning Page Boundary 3](#_Toc286066974)

[i. Incorrect Use of Literal 4](#_Toc286066975)

# 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 – Absolute Program

|  |  |
| --- | --- |
| 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:** | Matches Expected Output |

### Relative Program

|  |  |
| --- | --- |
| Assembly | Object File |
| RlTest .ORIG  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 <- x####  ST R0,Array ;M[Array] <-x####  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 | HRlTest00000018  T00000004  T00012200M0  T0002e007M0  T0003f022  T0004127f  T00050203M0  T0006480cM0  T00070068  T00080069  T00090021  T000a0020  T000b0000  T000c5020  T000d9000  T000e3013M0  T000fea13M0  T00102c17M0  T00117141  T0012f025  T00160010  T00170064  E0001 |
| **Result:** | Matches Expected Output |

### 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 |
| LasMem .ORIG xFFFF  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 | HLasMemffff0018  Tffff0004  T000023ff  T0001e006  T0002f022  T0003127f  T00040202  T0005480b  T00060068  T00070069  T00080021  T00090020  T000a0000  T000b5020  T000c9000  T000d3012  T000eea12  T000f2c16  T00107141  T0011f025  T00150010  T00160064  E0000 |
| **Result:** | Passes Test When it Should Not |

### Program Spanning Page Boundary

|  |  |
| --- | --- |
| Assembly | Object File |
| Lab2EG .ORIG xFFF7  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 <- xFFF7  ST R0,Array ;M[Array] <-xFFF7  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 | HSpnTstfff70018  Tfff70004  Tfff823f7  Tfff9e1fe  Tfffaf022  Tfffb127f  Tfffc03fa  Tfffd4803  Tfffe0068  Tffff0069  T00000021  T00010020  T00020000  T00035020  T00049000  T0005300a  T0006ea0a  T00072c0e  T00087141  T0009f025  T000d0010  T000e0064  Efff8 |
| **Result:** | Passes Test When it Should Not |

### Incorrect Use of Literal

|  |  |
| --- | --- |
| Assembly | Object File |
| Test03 .ORIG  Begin ADD R0, R1, =#100  .END Begin | Htest03.00000004  T00030064  E0000 |
| **Result:** | Program Doesn’t Complete Assembly |