# Lisa Pascal Compiler User Manual

## Introduction

Welcome to the Lisa Pascal Compiler! This user manual will guide you through the process of installing, building, and using the compiler to analyze and compile Pascal language programs. Follow the steps outlined in this manual to get started.

## Table of Contents

1. Installation

2. Compiling the Compiler

3. Running the Compiler

4. Understanding Error Messages

5. Example Usage

6. Frequently Asked Questions

## 1. Installation

To install the Lisa Pascal Compiler, you need the following tools installed on your system:

- Flex: A fast lexical analyzer generator.

- Bison: A general-purpose parser generator.

- GCC: The GNU Compiler Collection.

## 2. Compiling the Compiler

Follow these steps to compile the Lisa Pascal Compiler:

1. Open a terminal.

2. Navigate to the directory containing the Lisa Pascal Compiler source files (lisa.l and lisa.y).

3. Run the following commands to generate the lexical analyzer and parser:

```sh  
 flex lisa.l  
 bison -d lisa.y  
 ```

4. Compile the generated C files:

```sh  
 gcc -o lisa\_compiler lex.yy.c lisa.tab.c -lfl  
 ```

## 3. Running the Compiler

To execute the Lisa Pascal Compiler with a Pascal source file as input, use the following command:

```sh  
./lisa\_compiler source.pas  
```

Replace `source.pas` with the path to your Pascal source file.

## 4. Understanding Error Messages

The compiler provides error messages for different types of errors encountered during lexical analysis, syntax analysis, or semantic analysis:

- Lexer Errors: Unrecognized characters are reported with their location in the source file.

- Parser Errors: Syntax errors trigger an error message with the line number and a description of the issue.

- Semantic Errors: Duplicate entries, undeclared variables, and type mismatches result in error messages and terminate the compiler.

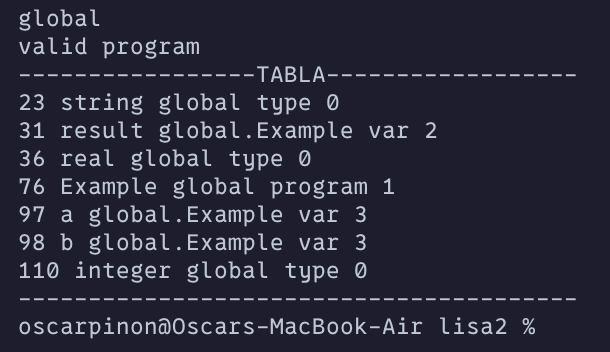
## 5. Example Usage

Here is an example of a simple Pascal program and how the Lisa Pascal Compiler processes it:

Example Pascal Program:

```pascal  
program Example;  
var  
 a, b, result: integer;  
begin  
 a := 5;  
 b := 10;  
 result := a + b;  
end.  
```

Running the Compiler:

```sh  
$ ./lisa\_compiler example.pas  
```

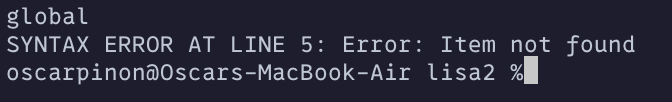
The compiler validates the program and outputs "valid program" if the syntax is correct.

ExampleIncorrect Pascal Program:

```pascal  
program ExampleIncorrect;  
var  
 b, result: integer; {notice a is never declared}  
begin  
 a := 5;  
 b := 10;  
 result := a + b;  
end.  
```

Running the Compiler:

```sh  
$ ./lisa\_compiler example.pas  
```



The compiler looks for identifiers in the table and outputs "Item not found" if the identifier doesn’t exist or is out of scope.