UNIVERSITY OF CENTRAL PUNJAB

Faculty of Information Technology

Project Phase 1

Submitted To: Ms.Aneela Mehmood
Name: Muhammad Usman
REG N0: L1F22BSCS1076
Section: G-10

Introduction:

This project is a lexical analyzer that reads a program file and identifies keywords, identifiers, numbers, operators, punctuations, and reports errors with line numbers. It helps understand how tokens in a program are recognized, which is a key step in compiler design.

1. Language Overview

Language Name:

P++ (Punjabi++)

Overview:

P++ is a Punjabi-inspired programming language that brings the logic and structure of C++ into a cultural and linguistic framework familiar to Punjabi speakers.
It uses Punjabi keywords and simple syntax to make coding feel natural, expressive, and inclusive.

Just like C++ extended C, P++ extends logic with culture making programming easier to understand for beginners who think in Punjabi.

Design Purpose:

The purpose of P++ is to eliminate the language barrier in learning programming.
By using Punjabi words instead of English keywords, students can:

- Understand programming logic faster
- Focus on reasoning, not memorization
- Feel comfortable learning in their mother tongue

Aim:

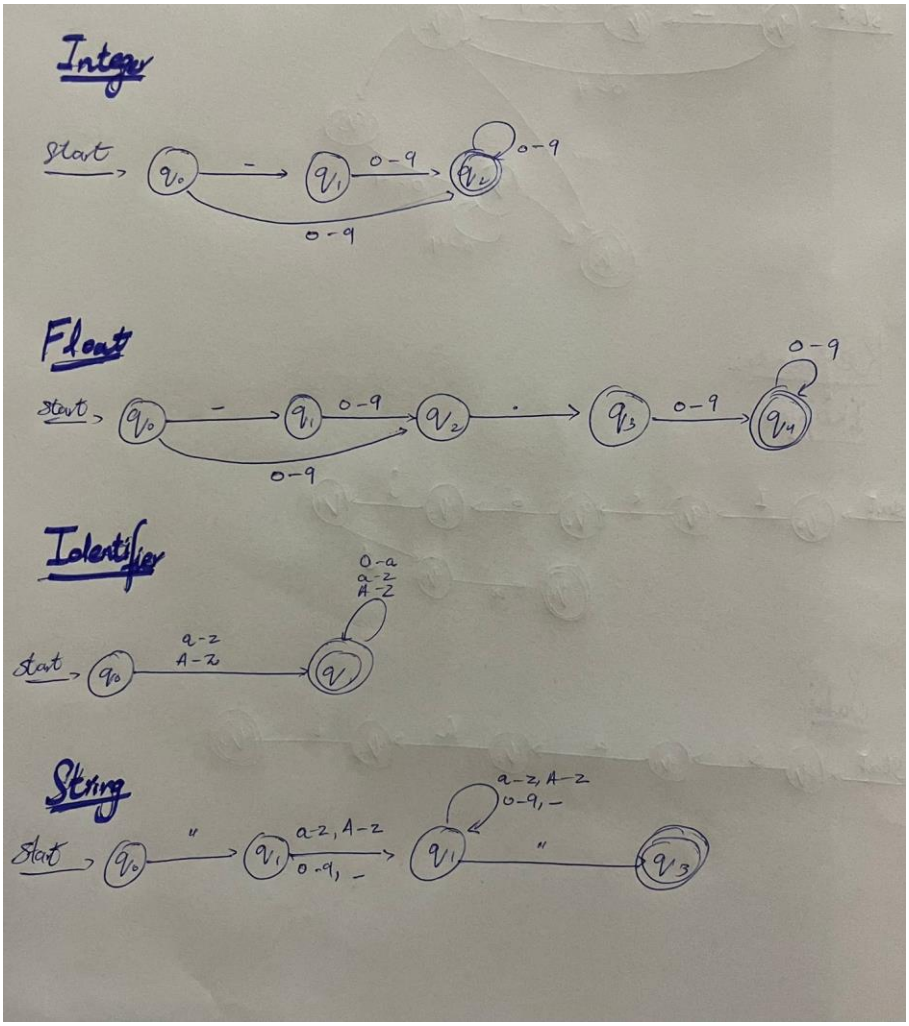
The main aims of **P++** are:

- Simplify programming logic using Punjabi syntax.
- Promote linguistic inclusion in computing education.
- Support education by introducing regional-language coding tools.
- Help learners transition smoothly from native logic to C++ or Java.
- Encourage creativity and confidence among non-English-speaking students.

Regex Table:

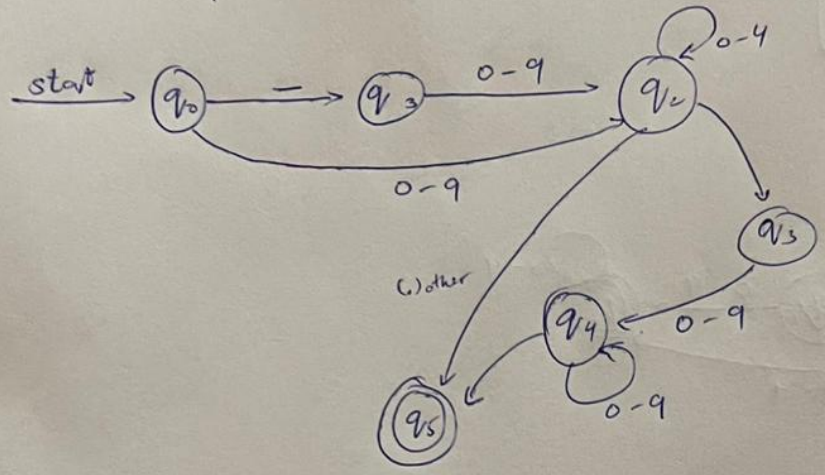
Token Type	Regular Expression (Regex)
Keyword	Je_tu | Nai_te | Ikojya |Diyo |Sahi | Ghalat | Lena | Wakhao | Rakam | Lafz | Ruk | Khokla | Yan |Aur| Ashariye
Identifier	[a-z A-Z _][a-z A-Z 0-9 _]*
Integer	(+|-)?[0-9]+ **OR** [0-9]+
Float	(+|-)?[0-9]+\(.)[0-9]+ **OR** [0-9]+\(.)[0-9]+
Exponential	[0-9]+(\.[0-9]+)?[eE][+|-]?[0-9]+
String	`"([^\"] **OR** (“)[a-z A-Z 0-9 (.)] (“)
Character	`'([^\'] **OR** (‘)[a-z A-Z 0-9 (.)] (‘)
Operator	"<-" | "=" | "==" | "!=" | "<=" | ">=" | "<" | ">" | "jorna" | "ghatana" | "zarab" | "taqseem" | "%"
Punctuation	"{", "}", "<<", ">>", ";", ",", ":"

Finite Automata:



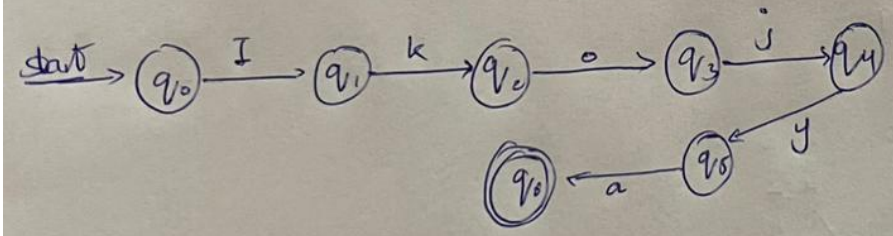
Number

(int + float)

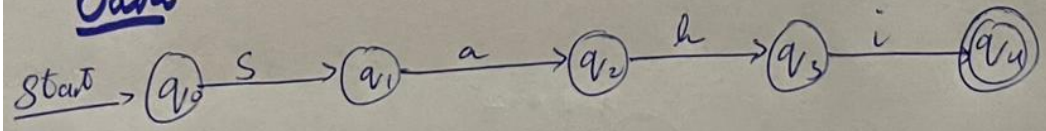


Keywords:-

Ikojya



Saki



Keywords:

Keyword	Meaning	Example (from code)	Explanation

Je_tu	if	Je_tu << !Ghalat >>
		{
		Wakhao "Negation works!";
		}
			Used to check a condition and execute a block of code if the condition is true.

Nai_te	else	Nai_te
		{
		Wakhao "Number is small";
		}
			Executes when the if condition is false.

Wakhao	cout / print / output	Wakhao "Number is greater than 5";	Displays text or variable values on the screen.
Lena	cin / input	Lena number;	Takes input from the user.

Ruk	break	Je_tu << i == 3 >>
		{
		Ruk; // break
		}
			Immediately exits from a loop or switch statement.

Diyo	return	Diyo 0;	Sends a value back from a function or ends its execution.
Rakam	int	Rakam x = 10;	Used to declare integer variables
Ashariye	float	Ashariye decimal=13.2;	Used to declare decimal or real number variables.
Sahi	true	Sahi truth = Sahi;	Represents a true logical value.
Ghalat	false	Ghalat lie = Ghalat;	Represents a false logical value.

Operators and Punctuations:

Category	Operator / Punctuation	Description
Operator	<-	Assign a value to a variable
Operator	=	Assign or compare values
Operator	==	Check if two values are equal
Operator	!=	Check if two values are not equal
Operator	<=	Less than or equal to
Operator	>=	Greater than or equal to
Operator	<	Less than
Operator	>	Greater than
Operator	jorna	Add two values
Operator	ghatana	Subtract one value from another
Operator	zarab	Multiply two values
Operator	taqseem	Divide one value by another
Operator	%	Get remainder of division
Punctuation	{	Start a block of code
Punctuation	}	End a block of code
Punctuation	<<	Start an expression or condition
Punctuation	>>	End an expression or condition
Punctuation	;	End a statement
Punctuation	,	Separate items or values
Punctuation	:	Used for special syntax or labels