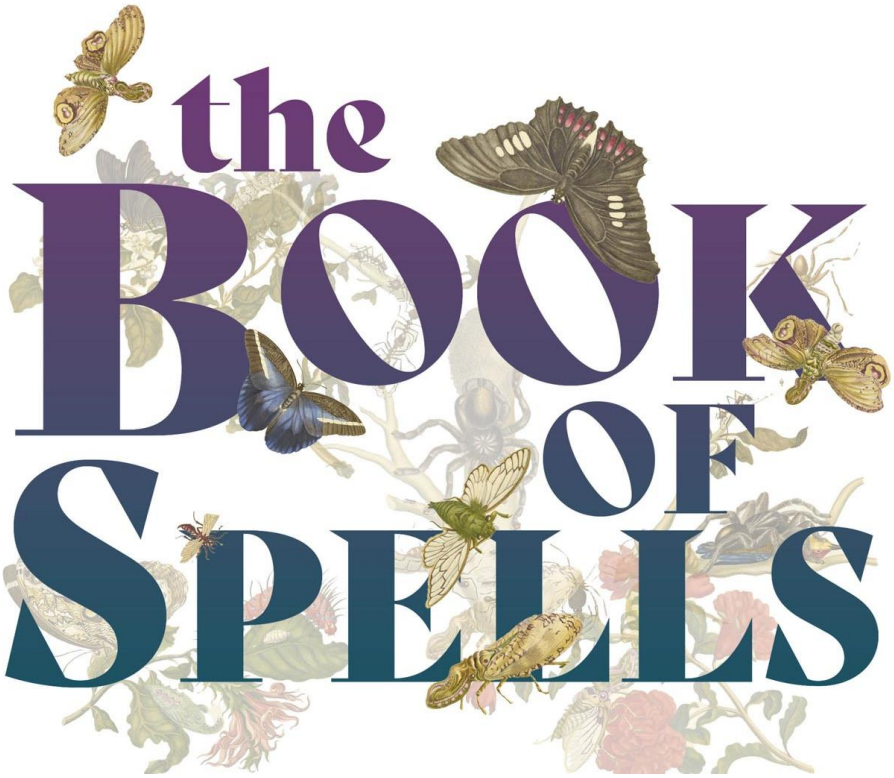
Compiler Construction



Project Phase 1 (Documentation)	
Submitted by	Muhammad Saad Sultan
Registration No.	L1F22BSCS0451
Section	G-1
Submitted to	Sir Asif Farooq
Project Topic	Lexical Analyzer (Tokenization, Error handling and Line Number track of each token)

Keywords:

Keyword	Meaning / Equivalent in C++	Example Usage
spell	Function or program start	`spell main { ... }`
summon	Declare a variable	`summon power = 10;`
cast	Print or output value	`cast "Fireball!";`
vanish	Delete or reset variable	`vanish energy;`
reveal	Return statement	`reveal result;`
bind	Assign value	`bind x = 5;`
Chant	Loop (for/while)	`chant (i < 5)`
until	While condition	`until (done == true)`
potion	Integer or float type	`potion mana = 100;`
magic	Boolean type	`magic isTrue = true;`
portal	If condition start	`portal (x > 0)`
elseportal	Else condition	`elseportal { ... }`
rune	String type	`rune name = "Saad";`
focus	Input value	`focus(x);`
endspell	End of program	`endspell;`

Operators:

Category	Symbol	Description
Operator	+	Adds two values
Operator	-	Difference of two values

Operator	*	Multiply two values
Operator	/	Divides two values
Operator	==	Compares two values
Operator	=	Used for assignment of values
Operator	>	Used for checking greater value
Operator	<	Used for checking smaller value
Operator	@	AND of two values
Operator	~	Not

Punctuations:

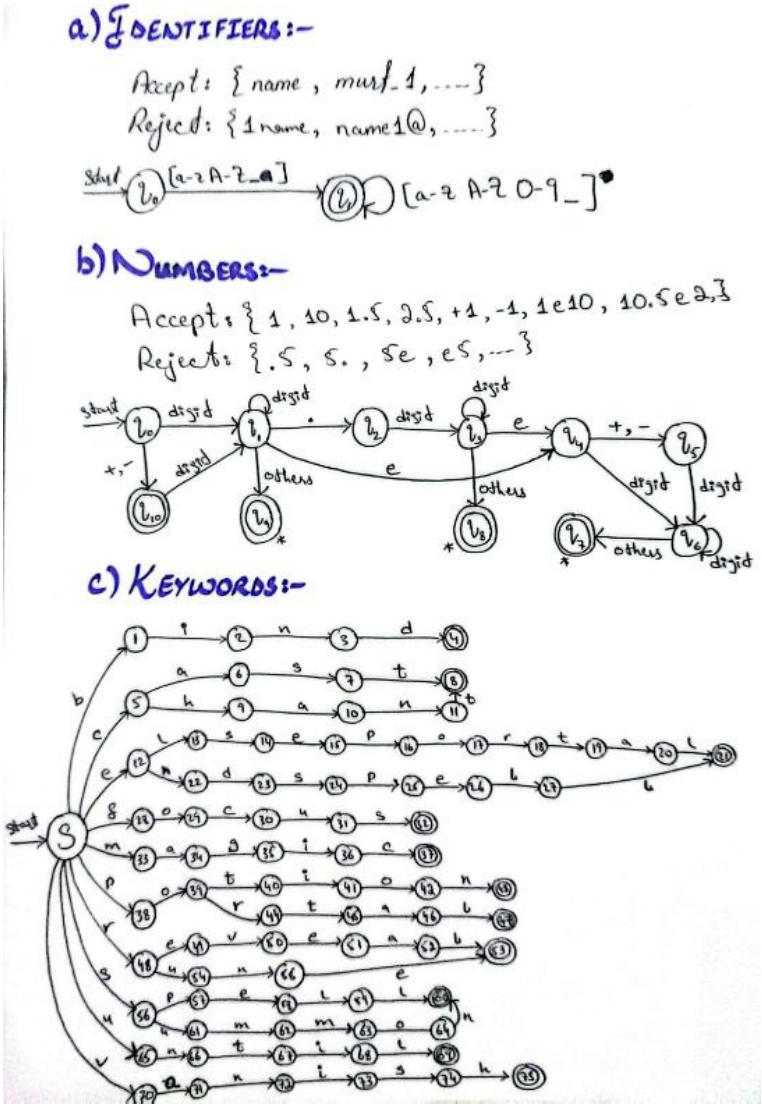
Category	Symbol	Description
Punctuation	;	End of a spell statement
Punctuation	{ }	Defines a block of spells
Punctuation	( )	For conditions or function calls
Punctuation	#	Single line comment
Punctuation	##…….##	Multi line comment

Regular Expressions (RE):

Token Type	Regular Expression	Example
Identifier	[a-z A-Z _][a-z A-Z 0-9 _]*	spell Name
Integer / Exponential	(+ | -)?[0-9]+(e(+|-)?[0-9]+)?	7150 / 7e2
Float	(+ | -)?[0-9]+\.[0-9]+(e(+|-)?[0-9]+)?	2.5 / 25e2
Keywords	[spell | summon | cast | reveal | bind | chant | until | potion | magic | portal elseportal | rune | focus | endspell]	cast “Abracadabra”;

Operators	[+ | - | * | / | == | = | > | < | @ | ~]	5+5
Punctuations	[; | { | } | (|) | #.* | ##.*##]	portal(power > 10)
String Literal	[\".*\"]	“Hocus Pocus!”

Finite Automata:



Explanation:

I created “SpellCode” to make programming more engaging and imaginative. The magical theme connects coding with the fantasy world, allowing beginners to understand programming logic through spell-like expressions.

Each keyword is designed to represent an action: summon brings something into existence (like variable declaration), and vanish removes it.

The structure remains close to C++, ensuring students can transition easily to real programming languages after learning SpellCode.