-MOHAMMAD

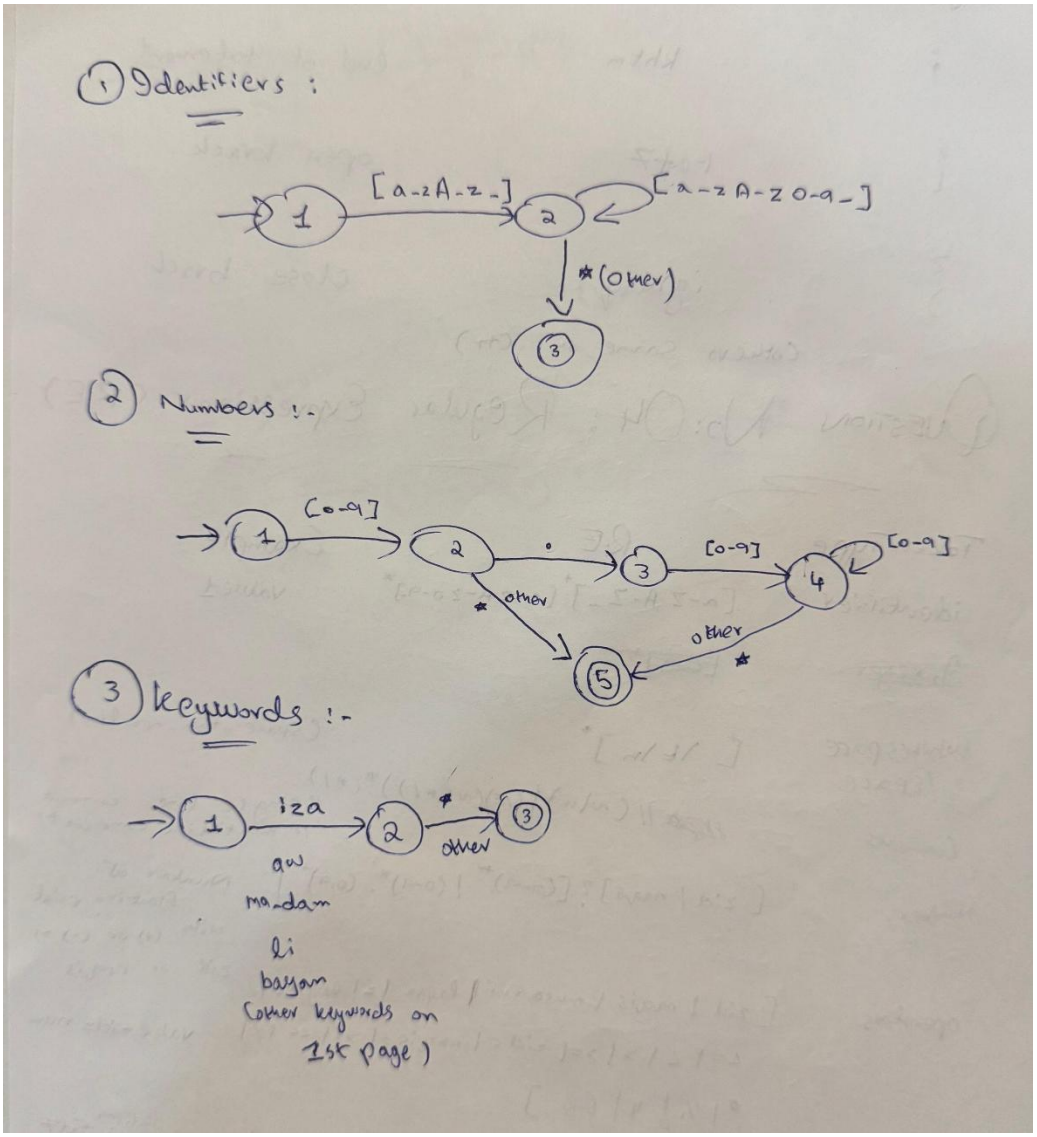
-L1F22BSCS0521

-G1 Compiler Construction

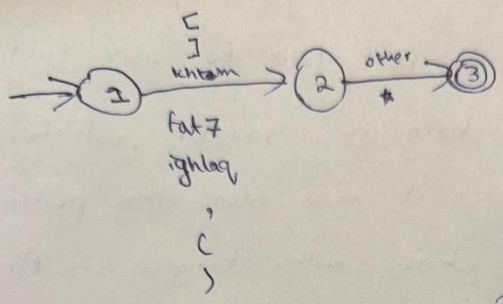
Regular Expression Table:

Token Type	Regular Expression
Identifier	[a-zA-Z_][a-zA-Z0-9_]*
Number (int/float/exp)	[zid | naqis]? [0-9]+(.[0-9]+)?([eE][zid|naqis]?[0-9]+)?
String	`"(\.
Character	`'([^'\]
Keyword	[bidaya | nihaya | adkhal | bayan | iza | aw | ma_dam | li | amal | raji3 | haqq | batil | Silsila | taifa | ruk |]
Operators	[zid | naqis | musaawi | laysa | = | laysa= | <= | >= | < | > | >> | << | naqis= | zid= | || | * | / | % | &&]
Punctuations	[khtam|fat7|ighlaq|!|,|]
Comment	//(^\n) | (/*)(^*/)*(*/)
Whitespace	[\t\r\n]+
Invalid Identifier	[0-9]+[a-zA-Z_][a-zA-Z0-9_]*

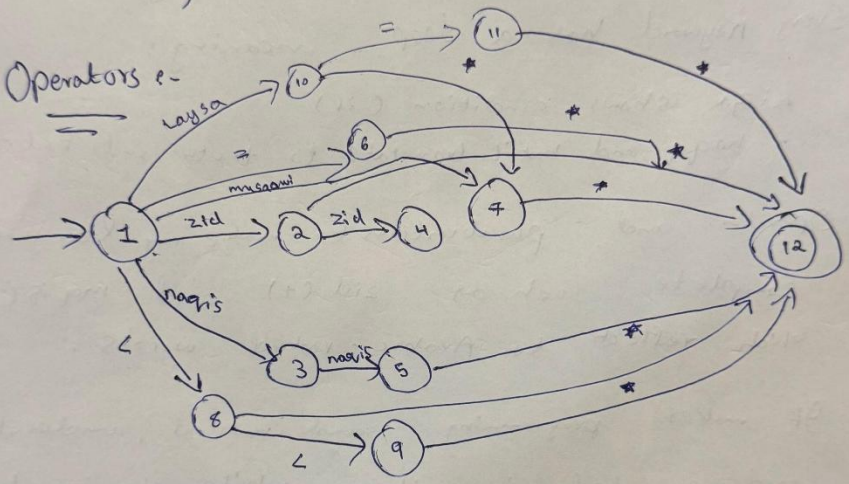
Finite Automata



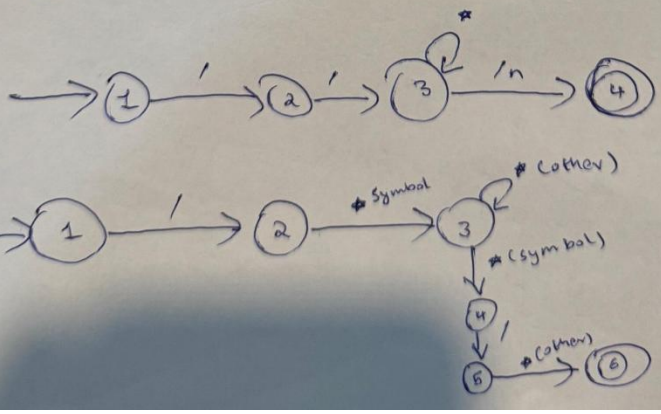
Punctuation :-



Operators :-



Comments :-



Keywords & Reason why

1. **bidaya** – Marks the start of the program. Similar to int main() in C/C++, giving a clear entry point.
2. **nihaya** – Marks the end of the program. Ensures the program terminates cleanly.
3. **adkhal** – Used to declare/input variables. Makes variable declaration explicit and readable.
4. **raji3** – Used to return a value from a function. Mirrors the return keyword in other languages.
5. **iza** – Represents an **if** statement. Provides conditional execution of code.
6. **aw** – Represents an else statement. Complements iza for alternate branches.
7. **li** – Represents a **for** loop. Chosen to iterate over ranges of values in a readable way.
8. **zid** – Represents **addition/increment** operator. Arabic-inspired keyword for arithmetic operations.
9. **naqis** – Represents **subtraction/decrement** operator. Complements zid for arithmetic flexibility.
10. **musaawi** – Represents **equality (==)** operator. Used for comparisons in conditional statements.
11. **laysa** – Represents **logical NOT (!)**. Allows negation in conditions.
12. **bayan** – Represents **output/print statement**. Chosen to provide program output functionality.
13. **haqq** – Represents **true** boolean value. Makes conditions more readable.
14. **batil** – Represents **false** boolean value. Complements haqq for logical operations.
15. **taifa** – Represents a **data structure or array declaration**.

All keywords are taken from Arabic, so it is easier for Arabic speakers to read and understand the code. Each keyword has its own clear purpose, like controlling the flow, doing math operations, printing output, or working with true/false values. This way, the code looks more organized and readable, and it helps students learn programming concepts in their own language without confusion.