Github link:

https://github.com/x64alex/Compilers/tree/main/Lab8

Run the program

Generate parser using Bison:
bison -d parser.y

Generate lexer using Flex:
flex scanner.lxi

Compile the parser and lexer with GCC:
gcc parser.tab.c

Run the generated executable with the specified input file (p3.cat):
./a.out p3.cat

Lexical Analysis (Lex File)

- **Regular Expression Definitions:**
  - Defines regular expressions for digits, non-zero digits, integer constants, letters, special characters, string constants, identifiers, and bad identifiers.
- **Rules Section:**
  - Maps reserved words to their respective tokens and prints information for each.
  - Maps operators to their respective tokens and prints information.
  - Maps separators to their respective tokens and prints information.
  - Recognizes identifiers and prints information.
  - Detects bad identifiers, prints an error message.
  - Recognizes integer constants and prints information.
  - Recognizes string constants and prints information.
  - Handles whitespace and newline characters.
  - Prints an error message for unrecognized tokens and exits.

Syntax Analysis (Yacc File)

The Yacc file (parser.y) specifies the syntax rules of the programming language and defines the parsing logic. It utilizes tokens generated by the Lex file.

Non-Terminals

- Main: The entry point for the parser, representing the entire program.
- CompoundStatement: Represents a block of statements enclosed in curly brackets.
- Statement: Represents various types of statements, such as declaration, assignment, if, while, and print statements.

- DeclarationStatement: Handles variable declarations, including arrays.
- Type: Represents variable types (int, str, arr).
- AssignmentStatement: Handles variable assignments.
- Expression: Represents mathematical expressions involving terms and factors.
- Term, Factor: Building blocks of expressions.
- ArrayStatement, ExpressionList: Handles array-related statements and expressions.
- IfStatement, WhileStatement, PrintStatement: Represents control flow and print statements.
- Condition: Represents conditional expressions.
- LogicCondition, LogicRelation: Extends conditionals to include logical operators (and, or).
- Relation: Represents relational operators (<, <=, ==, <>, >=, >).

Actions

- Each rule is associated with an action, printing information about the parsed structure.

Error Handling

- The yyerror function is defined to print an error message.