Programming 2023
Minimal BASIC Interpreter
DDL: Week13, Monday, 18:30

Introduction

In this assignment, your mission is to build a minimal BASIC interpreter. You may start with the code TA provided.

BASIC Language and Interpreter

The programming language BASIC - the name is an acronym for Beginner's All-purpose Symbolic Instruction Code - was developed in the mid-1960s at Dartmouth College by John Kemeny and Thomas Kurtz.

A Plus B

In BASIC, a program consists of a sequence of numbered statements, as illustrated by the simple program below:


```
Line number              A comment line
10 REM Program to add two numbers
20 INPUT n1
30 INPUT n2              Request value from user
40 LET total = n1 + n2
50 PRINT total          Assignment Statement
60 END                  Display value on console
    End of execution
```

Lexical

Identifiers are formed by one or more letters. Keywords that are reserved words in the language and cannot be used as identifiers. Integer literals are composed of digits only. **In this lab, the language is case-sensitive**, which means i and I are different variables, and IF, if, If, and even iF have different

meanings.

Line Numbers

The **line numbers** *at the beginning of the line* establish the sequence of operations in a program. In the absence of any control statements to the contrary, the statements in a program are **executed in ascending numerical order starting at the lowest number.**

Line numbers are also used to provide a simple editing mechanism. **Statements need not be entered in order,** because the line numbers indicate their relative position. Moreover, as long as the user has left gaps in the number sequence, new statements can be added in between other statements For example, to change the program that adds two numbers into one that adds three numbers, you would need to make the following changes:

```
                            ──── Add a new line, inserted into the
35 INPUT n3        program between line 30 and line 40
40 LET total = n1 + n2 + n3
   ╲ Replace the old line 40
```

The standard mechanism for deleting lines was to **type in a line number with nothing after it** on the line. Note that this operation actually **deleted the line** and did not simply replace it with a blank line that would appear in program listings.

Sequential Statements

REM	This statement is used for comments.
LET *var = exp*	This statement is BASIC's assignment statement.
PRINT *exp*	This statement print the value of the expression on the console and then **print a newline character**.
INPUT *var*	This statement print **a prompt consisting of the string " ? "** and then to read in a value to be stored in the variable.
END	Marks the end of the program. Execution halts when this line is reached. Execution **also stops if the program continues past the last numbered line**.

Control Statements

For example, the following BASIC program simulates a countdown from 10 to 0:

```
10  REM Program to simulate a countdown
20  LET T = 10 ───────── Jumps to line 70 if the
30  IF T < 0 THEN 70      result of comparison is true
40  PRINT T
50  LET T = T - 1
60  GOTO 30
70  END ──────── Jumps to line 30 unconditionally
```

GOTO *n*	This statement transfers control unconditionally to line n in the program. If line n **does not exist**, your BASIC interpreter should **generate an error message** informing the user of that fact.
IF *exp cmp exp* **THEN** *n*	This statement performs a conditional transfer of control. On encountering such a statement, the BASIC interpreter begins by evaluating condition, which in the minimal version of BASIC consists of two arithmetic expressions joined by one of the operators <, >, or =. Notice that **we use the same = symbol as in the assignment statmement**. If the result of the comparison is true, control passes to line n, **just as in the GOTO statement**; if not, the program continues with the next line in sequence.

Expressions

Expressions are used in LET, PRINT, and IF statements.

int_const	The simplest expressions are variables and integer constants.
var	
(*exp*)	
exp op exp	

Executed Directly

The LET, PRINT, and INPUT statements can be executed directly by typing them without a line number, in which case they are evaluated immediately. Thus, if you type in "PRINT 2 + 2" your program should ☐   ed   ☐   R e   execi   ☦N

CLEAR	This command deletes all program and variables.
QUIT	This command exits from the BASIC interpreter by calling exit(0).
HELP	This command provides a simple help message describing your interpreter.

Error Reporting

DIVIDE BY ZERO	Calculating some number divide by zero.
INVALID NUMBER	User types wrong value to answer INPUT statement.
VARIABLE NOT DEFINED	A variable used before assigned it.
LINE NUMBER ERROR	GOTO or IF statement's line number not exist.
SYNTAX ERROR	Any other errors.