Tiny BASIC

Chung-Yuan Huang

How to run Tiny BASIC

- Clone this program tb.py and run $ python3 tb.py.
- Of course, Python 3 is required to execute this software

Available Commands (1/3)

- REM <anything>
  - Used to write comments.
- LET <expression> = <expression>
  - Used to assign values to variables.
- RUN
  - Used to execute the stored code listing.
- INPUT <expression>
  - Used to accept user input into variables.

Available Commands (2/3)

- PRINT <expression>
  - Used to print values to the screen
- CLS
  - Used to clear the screen.
- GOTO <expression>
  - Used to jump the execution to another line.
- IF <expression> THEN <command>
  - Conditional statement.

Available Commands (3/3)

- END
  - Halts execution of a Tiny BASIC script.
- EXIT
  - Exits the Tiny BASIC interpreter.
- LIST
  - Prints the stored code listing to the screen.
- CLEAR
  - Clears the code listing.

Tiny BASIC Remarks (1/3)

- Tiny BASIC is a typed language.
- Numeric variable names may only contain characters from a-Z and numbers, provided they start with a letter.
- String variable names follow this same guidelines but must end in a $ character.
- For example, a variable called foo will be a numeric variable, while a variable called foo$ may store strings.

Tiny BASIC Remarks (2/3)

- Expressions work just like in most other programming languages you may know, with the only exception that values and operators must be separated with a space.
- For example a + b / 8 is a valid expression, while a+b/8 is not.
- Available operators
    - +, -, /, *, ^ (power), % (modulo), ==, !=, <, >, <=, >=, . (concatenation), & (logic and) and | (logic or).

Tiny BASIC Remarks (3/3)

- Lines written into the interpreter preceded by a line number (for example 10 PRINT "Hello there!") are not executed immediately, but added to the code listing.
- You may run all the lines in your code listing by using the run statement.
- Lines without a line number are executed right away.

Example LarBASIC Listings

```
10 REM +----------------------+
20 REM |Tiny BASIC Disan Count|
30 REM +----------------------+
40 PRINT "Enter a number:"
50 INPUT max
60 IF max % 2 == 0 THEN PRINT max . " is even!"
70 LET max = max - 1
80 IF max >= 0 THEN GOTO 60
90 PRINT "Done!”
100 END
```

Requirements (1/4)

- GOSUB / RETURN (20 points)
  - GOSUB is used to invoke a subroutine at the specified statement number.
  - RETURN is used, within the subroutine, to cause Tiny BASIC to pass control to the statement that immediately follows the GOSUB command that caused the subroutine to execute.
  - GOSUB / RETURN can be nested (10 points)
  - 100 GOSUB 200
  - 110 PRINT “HERE”
  - 200 PRINT ”Hello, World!”
  - 205 GOSUB 300
  - 210 RETURN
  - 300 PRINT “2nd”
  - 310 RETURN

Requirements (2/4)

- IF <expression> THEN <command> ELSE <command>
  - 5 points
- GOTOXY(row, column)
  - 5 points
- Extend operators
  - << (left shift), >> (right shift)
  - 5 points per operator

Requirements (3/4)

- SAVE <filename.tb>
  - 10 points
- LOAD <filename.tb>
  - 10 points
- FOR <variable> = <number> TO <number> DO <command>
  - The FOR command is used to set up execution loops.
  - 100 FOR X = 1 TO 100 DO PRINT “Hello, World!”
  - 20 points

  ```
  100 FOR I = 1 TO 10 STEP 2
  110 PRINT “Hello, World!”
  120 NEXT I (30~40 points)
  ```
- STOP
  - Used to stops the execution of a Tiny BASIC program and passes control to the Tiny BASIC monitor.

Requirements (4/4)

- STOP
  - Used to stops the execution of a Tiny BASIC program and passes control to the Tiny BASIC monitor.
  - 5 points
- Math functions
  - ABS(), SIN(), COS(), …
  - 5 points per function