TOBB University of Economics and Technology
Department of Computer Engineering
BIL395 Programming Languages
Instructor: Dr. Osman Abul

Assignment 1
Date due: February 19, 2023

Subject: Interpreter implementation for a text canvas painting language

Problem: In this assignment, you are expected to implement an interpreter for a text canvas painting language. The language is called $\mathcal{P}eakasso$ and its grammar description in **EBNF** is given below.

The language

<Peakasso> → PROGRAM <ID> ; <Canvas Init Section> <Brush Declaration Section> <Drawing Section>

<Canvas Init Section> → CANVAS-INIT-SECTION : <Canvas Size Init> <Cursor Pos Init>
<Canvas Size Init> → CONST CanvasX = *int_lit* ; CONST CanvasY = *int_lit* ;
<Cursor Pos Init> → CursorX = *int_lit* ; CursorY = *int_lit* ;

<Brush Declaration Section> → BRUSH-DECLARATION-SECTION : [<Variable Def>]
<Variable Def> → BRUSH <Brush List> ;
<Brush List> → <Brush Name> | <Brush Name>, <Brush List>
<Brush Name> → <ID> [= *int_lit int_lit*]
<ID> → *id*
<Drawing Section> → DRAWING-SECTION : { <Statement>;}
<Statement> → <Renew Stmt> | <Paint Stmt> | <Exhibit Stmt> | <Cursor Move Stmt>

<Renew Stmt> → RENEW-BRUSH '*message*' <Brush Name>
<Paint Stmt> → PAINT-CANVAS <Brush Name>
<Exhibit Stmt> → EXHIBIT-CANVAS
<Cursor Move Stmt> → MOVE <Cursor> TO <Expression>
<Cursor> → CursorX | CursorY

<Expression> → <Term> | <Expression> (PLUS | MINUS) <Term>
<Term> → <Factor>
<Factor> → *int_lit* | <Cursor> | CanvasX | CanvasY | (<Expression>)

Other key information about the language is as follows.

- Tokens: Other than the keyword lexemes, there are three tokens in the language. The first is *id* which is a letter followed by any number of letters or digits, *i.e.*, `["a"-"z","A"-"Z"] ( ["a"-"z","A"-"Z", "0"-"9"])*`. The second is the *int_lit* which is a sequence of digits between 0-9, with optional sign symbol (*e.g.*, "+" or "-"). The third one is the *message*, any sequence of characters other than newline, for user interaction.

- Comments: The rest of any line starting with the text "!!" is comment, therefore it should be skipped. Multi-line comments are not allowed.

- Whitespace characters: All whitespace characters, (*blank*, *tab* and *newline*), should be neglected. The only exception is the whitespace characters within the *message*.

- Variables and Types: Every variable belongs to the only type of BRUSH. Every variable (brush) has to be declared but need not to be explicitly initialized. Each brush has two components *height* and *width*, the default values for which are 1. Use of undeclared variables in statements is an error and this encounter terminates the program execution.

- Constants: CanvasX and CanvasY are constants, the value of which are initialized at the very beginning and remain the same until the program quits. The valid ranges are 5 to 200. In case an out of scope value is assigned at the initialization, give a warning and assume the respective value is 100.

- Operator Set: There are two arithmetic operators: addition and subtraction, the objective with which are to move the cursor on the canvas. The operators have the usual meanings for integer arithmetic. In the cursor move statement, in case the CursorX or CursorY moves out of the canvas, give a warning and use the old value, i.e., such movement statements are neglected.

- Operator Precedence: The operator precedences are already enforced by the grammar.

- Operator Associativity: The operator associativities are already enforced by the grammar.
- Ambiguity : The grammar is unambiguous but during implementation you may come up with some problems like lookahead. In such problematic cases, you need to rewrite some part of the grammar to get rid of the respective problem.
- Case sensitivity: The language is case-sensitive like C and Java. So, for instance, "A" and "a" are different entities.
- Reserved Keywords: All the lexemes (*e.g.*, RENEW-BRUSH, EXHIBIT-CANVAS, CursorX, CanvasX, PLUS, TO etc.) are reserved and can not be used as identifiers. If used, you are expected to raise a compile time error.
- Commands:
  - RENEW-BRUSH: The user is allowed to change the height and width of the respective brush.
  - PAINT-CANVAS: The respective brush stroke is applied on the canvas at the cursor location (CursorX, CursorY). Top-left of the brush starts at the cursor location.
  - EXHIBIT-CANVAS: The current painting is put on the exhibition, i.e., it is displayed on the screen.
  - MOVE: CursorX or CursorY is moved to a new location.

An example program

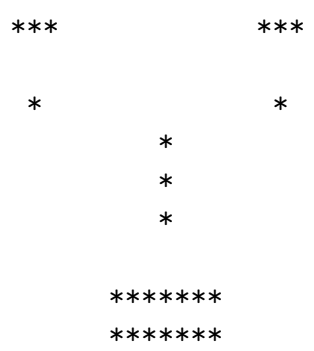
The source code given below is a valid *Peakasso* program to generate the !very art! given afterwards.

```
PROGRAM smiley;
CANVAS-INIT-SECTION :
!!  Define the canvas size and initial cursor location
CONST CanvasX = 50 ; CONST CanvasY = 50 ; CursorX = 1 ; CursorY = 1 ;

BRUSH-DECLARATION-SECTION : !!  Declare brushes
BRUSH goz = 1 2, agiz= 2 7, kas= 1 3, burun= 3 1 ;
```

```
DRAWING-SECTION : !!  Start drawing
RENEW-BRUSH 'eski goz degeri degeri en=1 boy=2.  Yenisini girin:'  goz;
!!  Assume the user enters 1 1
MOVE CursorX TO 5 ; MOVE CursorY TO CursorX MINUS 1 ;
PAINT-CANVAS kas;
MOVE CursorX TO CursorX PLUS 15 ;
PAINT-CANVAS kas;
MOVE CursorX TO 6 ; MOVE CursorY TO CursorY PLUS 2;
PAINT-CANVAS goz;
MOVE CursorX TO 21 ;
PAINT-CANVAS goz;
MOVE CursorX TO 7 ;
MOVE CursorY TO 7;
PAINT-CANVAS burun;
MOVE CursorX TO 11 ;
MOVE CursorY TO 11;
PAINT-CANVAS agiz;
EXHIBIT-CANVAS;
```

The exhibition looks like:



Implementation

Use *lex/yacc* (or *flex/Bison*). *Lex* is a lexical analyzer generator and *yacc* is a parser generator in C language.

To ease the development process, it is strongly advised that you first write a language recognizer and then fill the actions associated to each construct in a second round to obtain the fully functional interpreter. For incremental development, you may consider handling each expression notation in turn, and implementing the next one only after you are sure the previous one works properly.

References

- *lex/yacc* (or *flex/bison*) are standard tools in most Unix installations. If you are using Cygwin on Windows, these tools are not default but can be installed manually.

Delivery

Put all source files of your interpreter in a zip file and submit it through uzak.etu.edu.tr. You can ask help from the course assistant Oguzhan Canpolat at `aqwoguz@gmail.com`.

Important 1

Avoid cheating.

Important 2

You are allowed to work in pairs. Please do not ask for the triples or more.