NOMES: GEOVANNA CRISTINE CORRÊA RIBEIRO (202010137),

LUANA ROZA DE OLIVEIRA (202010258)

CURSO: CIÊNCIA DA COMPUTAÇÃO

DISCIPLINA: COMPILADORES

==================================================================

REGRAS DE PRODUÇÃO E PRIMITIVAS:

A: FO num {turtle.draw\_segment ( 0, $num.value)}

B: BK num {turtle.draw\_segment ( 180, $num.value)}

C: LT angle {turtle.draw\_segment ( $angle.value, $1)}

D: RT angle {turtle.draw\_segment ( $angle.value, $1)}

E: PD {turtle.set\_draw (T)}

F: PU {turtle.set\_draw (F)}

G: WC {turtle.clear()}

H: CS {turtle.clear()}

I: SETXY num num {turtle.set\_position ($num.value, $num.value)}

J: HOME {turtle.home()}

K: XCOR {turtle.xcor}

L: YCOR {turtle.ycor}

M: HEADING {$angle.value, $0}

N: IF {turtle.set\_IF(T)}

O: IF\_ELSE {turtle.set\_IF(T) or turtle.set\_ELSE(F)}

P: WHILE: {turtle.end()}

Q: PRINT {turtle.date()}

R: TYPEIN {turtle.read()}

S: TO {turtle.new\_primitive}

T: RANDOM {turtle.num\_integer($0 a $9)}

TO SQUARE :num

PD {turtle.set\_draw (T)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 90 {turtle.draw\_segment ( 90, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 90 {turtle.draw\_segment ( 90, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 90 {turtle.draw\_segment ( 90, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 90 {turtle.draw\_segment ( 90, $1)}

PU {turtle.set\_draw (F)}

END

TO HEX :num

PD {turtle.set\_draw (T)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 60 {turtle.draw\_segment ( 60, $1)}

PU {turtle.set\_draw (F)}

END

TO TRIANGLE :num

PD {turtle.set\_draw (T)}

RT 60 {turtle.draw\_segment ( 60, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 120 {turtle.draw\_segment ( 120, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

RT 120 {turtle.draw\_segment ( 120, $1)}

FO :num {turtle.draw\_segment ( 0, :$num.value)}

PU {turtle.set\_draw (F)}

END

=================================================================

program A:

SETXY 20 0 {turtle.set\_position (20, 0)}

PD {turtle.set\_draw (T)}

BK 50 {turtle.draw\_segment (180, 50)}

PU {turtle.set\_draw (F)}

FO 100 {turtle.draw\_segment (0, 100)}

PD {turtle.set\_draw (T)}

FO 50 {turtle.draw\_segment (0, 50)}

PU {turtle.set\_draw (F)}

SETXY 40 0 {turtle.set\_position (40, 0)}

PD {turtle.set\_draw (T)}

BK 50 {turtle.draw\_segment (180, 50)}

PU {turtle.set\_draw (F)}

FO 100 {turtle.draw\_segment (0, 100)}

PD {turtle.set\_draw (T)}

FO 50 {turtle.draw\_segment (0, 50)}

PU {turtle.set\_draw (F)}

SETXY 30 25 {turtle.set\_position (30, 25)}

PD {turtle.set\_draw (T)}

SQUARE 50

LT 60

SQUARE 50

LT 60

SQUARE 50

LT 60 // repeat 6 [

SQUARE 50 // SQUARE 50

LT 60 // LT 60

SQUARE 50 // ]

LT 60

SQUARE 50

LT 60

PU {turtle.set\_draw (F)}

END

CS {turtle.clear()}

program B:

PD {turtle.set\_draw (T)}

HEX 60

PU {turtle.set\_draw (F)}

END

WC {turtle.clear ()}

program C:

PD {turtle.set\_draw (T)}

SQUARE 200

SETXY 100 100 {turtle.set\_position (100, 100)}

TRIANGLE 150

RT 60

TRIANGLE 150 // repeat 4 [

RT 60 // TRIANGLE 150

TRIANGLE 150 // RT 60

RT 60 // ]

TRIANGLE 150

RT 60

PU {turtle.set\_draw (F)}

END

CS {turtle.clear ()}

program D:

HEADING {$angle.value, $0}

SETXY 30 0 {turtle.set\_position (30, 0)}

FO 100 {turtle.draw\_segment ( 100, draw\_segment(0, 100)}

PD {turtle.set\_draw (T)}

SQUARE 200

TRIANGLE 120

RT 60

PU {turtle.set\_draw (F)}

HOME {turtle.home()}

program E:

PD {turtle.set\_draw (T)}

SQUARE 100

SETXY 50 50 {turtle.set\_position (50, 50)}

FO 30 {turtle.draw\_segment (0, 30)}

RT 60

BK 40 {turtle.draw\_segment (180, 20)}

IF\_ELSE {turtle.set\_IF(T) or turtle.set\_ELSE(F)}

RANDOM {turtle.num\_integer($9)}

TYPEIN {turtle.read()}

PRINT {turtle.date()}