```
/*
   Homework_03: Formal languages
   BNF and EBNF notation

   ITESM QRO
   Edgar Yair Camacho Paredes - A01206849
   11/07/2022
*/

// BNF for switch statement in C
<switch>  ::= switch(<expresion>)<nl><indent>{<case_values>}
<expresion> ::= <string>
                | <number>
<case_values> ::= <case_value>:<statements>; break;
                  | <case_value>:<statements>; break;<nl><case_values>
<case_value>  ::= <string>
                  | <number>
                  | default
<statements>  ::= <statement>
                  | <statement><statements>
<statement> ::= <some_code>
<nl>  ::= \n
<indent>  ::= \t

// BNF for "for" loop in Python
<for> ::= for <iterating_var> in <sequence>: <nl><indent><statements>
<iterating_var> ::= <sequence_item>
<sequence>  ::= <sequence_item>
                | <sequence_item><sequence_item>
<sequence_item> ::= <character>
                    | <string>
                    | <number>
<nl>  ::= \n
<indent>  ::= \t
<statements>  ::= <statement>
                  | <statement><statements>
<statement> ::= <some_code>

// BNF for case statement in Racket
<case>  ::= (case <valuable_expresion> [<case_clause> <statements>])
<valuable_expresion>  ::= <expresion>
<expresion> ::= (<expresion>)
                | <function>
                | <number>
                | <string>
                | <symbol>
<case_clause> ::= (<list>)
<list>  ::= <list_element>
```

```
         | <list_element><list>
<list_element>  ::= <character>
                  | <number>
                  | <string>
<statements>  ::= <statement>
                 | <statement><statements>
<statement> ::= <some_code>
```