```
PROGRAM
        : "start" STATEMENTS "slut"

STATEMENTS
        : STATEMENT
        | STATEMENTS STATEMENT

STATEMENT
         : ARRAY_ADD_ELEMENT
        | ARRAY_REMOVE_BY_INDEX
        | ARRAY_REMOVE_BY_VALUE
        | RETURN
        | FUNCTION
        | FUNCTION_CALL
        | BREAK
        | PRINT
        | IF_BLOCK
        | WHILE_LOOP
        | FOR_LOOP
        | ASSIGN
        | START_ASSIGN

PRINT
         : "skriv" ARRAY_INDEXING
        | "skriv" STRING_ADDING
        | "skriv" STRING_EXPR
        | "skriv" EXPR)

IDENTIFIER
        : /[a-zA-ZåäöÅÄÖ]+[a-zA-ZåäöÅÄÖ0-9_]*/

FUNCTION
        : "inget" "funktion" IDENTIFIER "(" PARAMETER_NAME ")" STATEMENTS "slut"/
        | "inget" "funktion" IDENTIFIER "(" ")" STATEMENTS "slut"/
        | "tal" "funktion" IDENTIFIER "(" PARAMETER_NAME ")" STATEMENTS"slut"/
        | "tal" "funktion" IDENTIFIER "(" ")" STATEMENTS"slut"/
        | "strängar" "funktion" IDENTIFIER "(" PARAMETER_NAME ")" STATEMENTS
"slut"/
        | "strängar" "funktion" IDENTIFIER "(" ")" STATEMENTS"slut"/

FUNCTION_CALL
        : IDENTIFIER "(" ")"
        | IDENTIFIER "(" PARAMETERS ")"

RETURN
        : "returnera" EXPR
        | "returnera" STRING_EXPR

PARAMETER_NAME
        : IDENTIFIER
        | PARAMETER_NAME "," IDENTIFIER

PARAMETERS
        : PARAMETER
        | PARAMETERS "," PARAMETER

PARAMETER
        : EXPR
```

```
        | IDENTIFIER

WHILE_LOOP
        : "medans" "(" BOOL_TEST ")" STATEMENTS "slut"

FOR_LOOP
        : "för" "(" BOOL_TEST ASSIGN ")" STATEMENTS "slut"

BREAK
        : "avbryt"

TERM
        : TERM "*" ATOM
        | TERM "/" ATOM
        | TERM "%" ATOM
        | FUNCTION_CALL
        | ATOM

EXPR
        : EXPR "+" TERM
        | EXPR "-" TERM
        | TERM

STRING_ADDING
        :STRING_EXPR "&" STRING_EXPR
        |:STRING_ADDING "&" STRING_EXPR

IF_BLOCK
        : IF_RULE

IF_RULE
        : IF ELSE_IF ELSE_ "slut"
        | IF ELSE_ "slut"
        | IF ELSE_IF "slut"
        | IF "slut"

IF
        : "om" "(" BOOLEAN_EXPRESSION, ")" STATEMENTS

ELSE_IF
        : "annars_om" "(" BOOLEAN_EXPRESSION ")" STATEMENTS
        | ELSE_IF ELSE_IF

ELSE
        :"annars" STATEMENTS

START_ASSIGN
        : "sträng" IDENTIFIER "=" STRING_ADDING
        | "tal" IDENTIFIER "=" EXPR
        | "bool" IDENTIFIER "=" BOOLEAN_EXPRESSION
        | "sträng" IDENTIFIER "=" STRING_EXPR
        | "lista" IDENTIFIER "=" ARRAY

ASSIGN
        : IDENTIFIER "=" STRING_ADDING
        | IDENTIFIER "=" EXPR
        | IDENTIFIER "=" BOOLEAN_EXPRESSION
        | IDENTIFIER "=" STRING_EXPR
```

```
        | IDENTIFIER "=" ARRAY

BOOLEAN_EXPRESSION
        :EXPR "<" EXPR
        | EXPR ">" EXPR
        | EXPR "<=" EXPR
        | EXPR ">=" EXPR
        | EXPR "!=" EXPR
        | EXPR "==" EXPR
        | BOOLEAN_EXPRESSION "och" BOOLEAN_EXPRESSION
        | BOOLEAN_EXPRESSION "eller" BOOLEAN_EXPRESSION
        | BOOLEAN_EXPRESSION "antingen_eller" BOOLEAN_EXPRESSION
        | "inte" BOOLEAN_EXPRESSION

ARRAY_SIZE
        : "storlek_på" IDENTIFIER

ARRAY_INDEXING
        : IDENTIFIER "[" EXPR "]"

ARRAY_ADD_ELEMENT
        : IDENTIFIER "lägg_till" STRING_EXPR
        | IDENTIFIER "lägg_till" EXPR

ARRAY_REMOVE_BY_INDEX
        : IDENTIFIER "ta_bort_index" EXPR

ARRAY_REMOVE_BY_VALUE
        : IDENTIFIER "ta_bort_värde" STRING_EXPR
        | IDENTIFIER "ta_bort_värde" EXPR

ARRAY
        : "[" ARRAY_VALUES "]"

ARRAY_VALUES
        : STRING_EXPR
        | ARRAY_VALUES "," STRING_EXPR
        | EXPR
        | ARRAY_VALUES "," EXPR

STRING_EXPR
        : FUNCTION_CALL
        | /"[^\"]*"/
        | /'[^\']*'/
        | IDENTIFIER

ATOM
        : ARRAY_INDEXING
        | ARRAY_SIZE
        | FUNCTION_CALL
        | Float
        | Integer
        | "(" EXPR ")"
        | IDENTIFIER
```