

Ion Iulia-Andreea

<https://github.com/iuliaaion>

A) Language specification

1. Language Definition

1. Alphabet

- a. upper(A-Z) and lower case letters(a-z) of the Roumanian alphabet
- b. underline character '_'
- c. decimal digits(0-9)

Lexic

a. Special symbols

- operators + - * / = < <= == => >
- separators [] {} : ; space
- reserved words: ARRAY, CONST, CHAR, DO, ELSE, IF, INT, OF, PROGRAM, READ, THEN, WHILE, WRITE, DOUBLE

b. Identifiers

- a sequence of letters and digits s.t. the first character is a letter
 $\text{letter} \langle \text{letter} \rangle ::= \text{"A"} \mid \text{"B"} \mid \dots \mid \text{"Z"}$
 $\langle \text{digit} \rangle ::= \text{"0"} \mid \text{"1"} \mid \dots \mid \text{"9"}$
 $\langle \text{letter_or_digit} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{digit} \rangle$
 $\langle \text{id} \rangle ::= \langle \text{letter_or_digit} \rangle \mid \langle \text{letter_or_digit} \rangle \langle \text{id} \rangle$
id = identifier

c. Constants

1. integer

- $\langle \text{no_const} \rangle ::= + \langle \text{nr} \rangle \mid - \langle \text{nr} \rangle \mid \langle \text{nr} \rangle$
- $\langle \text{nr} \rangle ::= \langle \text{no_zero} \rangle \mid \langle \text{no_zero} \rangle \langle \text{digit_seq} \rangle$
- $\langle \text{no_zero} \rangle ::= \text{"1"} \mid \text{"2"} \mid \dots \mid \text{"9"}$
- $\langle \text{digit_seq} \rangle ::= \langle \text{digit} \rangle \mid \langle \text{digit} \rangle \langle \text{digit_seq} \rangle$

2.character

$\langle \text{character} \rangle ::= \langle \text{letter} \rangle \mid \langle \text{digit} \rangle$

3.string

$\langle \text{string} \rangle ::= \langle \text{character} \rangle \langle \text{string} \rangle$

2.Syntax

-the words that are predefined tokens are specified between " "

a.Syntactical rules

$\langle \text{id_sequence} \rangle ::= \langle \text{id} \rangle \mid \langle \text{id} \rangle \text{ " , " } \langle \text{id_sequence} \rangle$

$\langle \text{decl} \rangle ::= \langle \text{id_sequence} \rangle \text{ " " } \langle \text{type} \rangle$

$\langle \text{type} \rangle ::= \langle \text{typo} \rangle \mid \langle \text{decl_arr} \rangle$

$\langle \text{decl_arr} \rangle ::= \text{ "ARRAY" " [" } \langle \text{nr} \rangle \text{ "] " "OF" } \langle \text{typo} \rangle$

$\langle \text{typo} \rangle ::= \text{ "bool" } \mid \text{ "char" } \mid \text{ "int" } \mid \text{ "double" }$

$\langle \text{cmpd_stmt} \rangle ::= \text{ " { " } } \langle \text{stmtlist} \rangle \text{ " } \text{ " }$

$\langle \text{stmt_list} \rangle ::= \langle \text{stmt} \rangle \mid \langle \text{stmt} \rangle \text{ " ; " } \langle \text{stmt_list} \rangle$

$\langle \text{stmt} \rangle ::= \langle \text{simple_stmt} \rangle \mid \langle \text{struct_stmt} \rangle$

$\langle \text{simple_stmt} \rangle ::= \langle \text{assign_stmt} \rangle \mid \langle \text{io_stmt} \rangle$

$\langle \text{assign_stmt} \rangle ::= \langle \text{id} \rangle \text{ " = " } \langle \text{expr} \rangle$

$\langle \text{expr} \rangle ::= \langle \text{expr} \rangle \text{ " + " } \langle \text{term} \rangle \mid \langle \text{expr} \rangle \text{ " - " } \langle \text{term} \rangle \mid \langle \text{term} \rangle$

$\langle \text{term} \rangle ::= \langle \text{term} \rangle \text{ " * " } \langle \text{factor} \rangle \mid \langle \text{term} \rangle \text{ " / " } \langle \text{factor} \rangle \mid \langle \text{factor} \rangle$

$\langle \text{factor} \rangle ::= \text{ " (" } \langle \text{expr} \rangle \text{ ") " } \mid \langle \text{id} \rangle$

$\langle \text{io_stmt} \rangle ::= \text{ "READ" " " (" } \langle \text{id} \rangle \text{ ") " } \mid \text{ "WRITE" " " (" } \langle \text{id} \rangle \text{ ") " }$

$\langle \text{struct_stmt} \rangle ::= \langle \text{cmpd_stmt} \rangle \mid \langle \text{if_stmt} \rangle \mid \langle \text{while_stmt} \rangle \mid \langle \text{for_stmt} \rangle$

$\langle \text{if_stmt} \rangle ::= \text{ "IF" } \langle \text{cond} \rangle \text{ "THEN" } \langle \text{stmt} \rangle \text{ ["ELSE" } \langle \text{stmt} \rangle \text{] }$

$\langle \text{while_stmt} \rangle ::= \text{ "WHILE" } \langle \text{cond} \rangle \text{ "DO" } \langle \text{stmt} \rangle$

$\langle \text{for_stmt} \rangle ::= \text{ "FOR" } \langle \text{cond} \rangle \text{ "TO" } \langle \text{nr} \rangle \mid \langle \text{id} \rangle \text{ "DO" } \langle \text{stmt} \rangle$

$\langle \text{cond} \rangle ::= \langle \text{expr} \rangle \langle \text{relation} \rangle \langle \text{expr} \rangle$

$\langle \text{relation} \rangle ::= \text{ " < " } \mid \text{ " < = " } \mid \text{ " = = " } \mid \text{ " ! = " } \mid \text{ " > = " } \mid \text{ " > " }$

B) Sum of all prime numbers less than k

```
int i, j, k, n, prim,sum;
READ(k);
FOR i=2 TO k DO
{
    sum=0;
    prim=1;
    n=i/2;
    FOR j=2 TO n DO
    {
        IF i%j==0 THEN
            { prim=0; }
        IF prim=1 THEN
            { WRITE(i);
              sum=sum+i; }
    }
}
```