**Analysis of Programming Language IAG0450 Exam 2015(50p,with test)**

**A.Theoretical part 25p**

**Syntax diagram and description in metalanguage**

**The grammar**

**Important part of language design**

**Type of translators**

**Translation and interpretation**

**Multi-level languages**

**Evaluation of programming language**

**Desirable characteristics**

**Essential characteristics**

**The scanner**

**The parser**

**The code generator and optimization**

**Error handling**

**General expressions**

**Assigment statements**

**IF statement**

**GO-TO statement**

**The loop statements**

**Data types and variables**

**Meaning of parentheses**

**The stack as a programming tools**

**Structured programming**

**Subroutines**

**Built-in functions**

**User-defined functions**

[**http://groups.engin.umd.umich.edu/CIS/course.des/cis400/**](http://groups.engin.umd.umich.edu/CIS/course.des/cis400/)

**B.Praxis part 25p**

1.Write Flex rules for( define token) INT, DOUBLE,VAR and yylval:

**%union{**

**int intval;**

**double doubleval;**

**char \*name;**

**}**

2.We have Flex rules:

**digit [0-9]**

**alpha [a-zA-Z]**

**alphanum [0-9a-zA-Z]**

**%%**

**"PROCEDURE" { return PROC; }**

**"PRINT" { return PRINT;}**

**{digit}+ { /\*täidetakse yylval\*/ return INT;}**

**{alpha}{alphanum}\* { /\*täidetakse yylval\*/ return VAR;}**

**[=;{}()+-/\*] { return yytext[0];}**

**[ \t\n] {;}**

and source

**PROCEDURE Arvuta ( x )**

**{**

**y = x \* x - 5;**

**PRINT y;**

**}**

Write and explane the tokens sequence for this.

3.We have token VAR and Bison rules:

**vars: VAR { $$ = push(id($1)); free($1); }**

**| VAR vars { $$ = push(id($1)); free($1); }**

**;**

and source **a f c g**

Descibe translater work.