Lab 1   Nedelcu Dania

```
#include <iostream>

using namespace std;

  int factorial(int n) {

        int fact = 1;

        for (int i = 1; i <= n; i++) {

                fact = fact * i;

        }

        return fact;

}


int main() {

        int n;

        cout << "Enter a positive integer: ";

        cin >> n;


        if (n < 0) {

                cout << "Factorial is not defined for negative numbers." << endl;

        } else {

                int result = factorial(n);

                cout << "Factorial of " << n << " is: " << result << endl;

        }
        return 0;

}
```

a.) program ::= declarationList ";" cmpdstmt "."

declarationList ::= declaration | declaration ";" declarationList

declaration ::= identifier ":" type

type ::= "int" | "char" | "boolean" | "struct"

cmpdstmt ::= "{" stmtlist "}"

stmtlist ::= stmt | stmt ";" stmtlist

stmt ::= simplstmt | ifstmt | forstmt | functionStatement

simplstmt ::= assignstmt | iostmt

assignstmt ::= identifier ":=" expression

iostmt ::= "cin" ">>" identifier | "cout" "<<" identifier

ifstmt ::= "if" condition "{" stmtlist "}" ["else" "{" stmtlist "}"]

forstmt ::= "for" "(" assignstmt ";" condition ";" assignstmt ")" "{" stmtlist "}"

functionStatement ::= type identifier "(" argumentsList ")" "{" stmtlist "return" expression ";" "}"

argumentsList ::= identifier ":" type | identifier ":" type "," argumentsList

expression ::= expression "+" term | expression "-" term | term

term ::= term "*" factor | term "/" factor | factor

factor ::= identifier | const | "(" expression ")"

condition ::= expression relation expression

relation ::= ">" | "<" | "=" | "<>" | ">=" | "<="

identifier ::= letter (letter | digit)*

letter ::= "A" | "B" | ... | "Z" | "a" | "b" | ... | "z"

digit ::= "0" | "1" | "2" | "3" | "4" | "5" | "6" | "7" | "8" | "9"

```
const ::= number | character

number ::= digit+

character ::= "'" letter "'"
```

```
functionStatement ::= "int" "factorial" "(" "n" ":" "int" ")"
                      "{" "fact := 1" ";"
                      "for" "(" "i := 1" ";" "i <= n" ";" "i := i + 1" ")"
                      "{" "fact := fact * i" ";" "}"
                      "return" "fact" ";" "}"
```

```
ifstmt ::= "if" "n < 0" "{" "cout << 'Factorial is not defined for negative numbers.'" "}"

iostmt ::= "cin >> n" | "cout << 'Factorial of' << n << 'is:' << result"
```

```
b.) function factorial:int n {
    int fact = 1
    int i
    for   i = 1; i <= n; i++
        fact = fact * i

    return fact
}

function main {
    int n
    print "Enter a positive integer: "
    input n
    if n < 0
        print   "Factorial is not defined for negative numbers."
        else
            int result = factorial(n)
            print "Factorial of "+ n+ " is: "+ result ""
    return 0
}
```

language rules:

function : defines a function, should be followed by the name of the function and {} and if there are arguments use ":"type of variable" "argument""

there are no parantheses so instead of them for the "if", "for", use tab

there is no ";" after a condition, statement

to display a message use print" "

type of variables: int, float, string, bool, char

operators: <= == >= := + - = * / % and or