# Project 1: Scanner

## 1-Input:

Multiple lines of code written in TINY language syntax

## 2-Output:

List of ( Lexeme, tokenvalue)

Example:

| | |
|---|---|
| x | Id_x |
| := | T_assign operator |
| 4 | T_number |

## 3- Deliverables:

1- Executable
2- GUI Desktop progrem
3- Code files

## 4-Deadline:

Sunday 5/4/2020

## 5-Rule of Tiny Language described as:

1) Number: any sequence of digits and maybe floats (e.g. 123 | 554 | 205 | 0.23 | …)

2) String: starts with double quotes followed by any combination of characters and digits then ends with double quotes (e.g. “Hello” | “2nd + 3rd” | …)

3) Reserved_Keywords: int | float | string | read | write | repeat | until | if | elseif | else | then | return | endl

4) Comment_Statement: starts with /* followed by any combination of characters and digits then ends with */ (e.g. /*this is a comment*/ | …)

5) Identifiers: starts with letter then any combination of letters and digits. (e.g. x | val | counter1 | str1 | s2 | …)

6) Function_Call: starts with Identifier then left bracket “(“ followed by zero or more Identifier separated by “,” and ends with right bracket “)”. (e.g. sum(a,b) | factorial(c) | rand() | … )

7) Arithmatic_Operator: any arithmetic operation (+ | - | * | / )

8) Assignment_Statement: starts with Identifier then assignment operator “:=” followed by Expression (e.g. x := 1 | y:= 2+3 | z := 2+3*2+(2-3)/1 | …)

9) Datatype: set of reserved keywords (int, float, string)

10) Write_Statement: starts with reserved keyword “write” followed by an Expression or endl and ends with semi-colon (e.g. write x; | write 5; | write 3+5; | write “Hello World”; | …)
11) Read_Statement: starts with reserved keyword “read” followed by an Identifier and ends with semi-colon (e.g. read x; | …)

12) Return_Statement: starts with reserved keyword “return” followed by Expression then ends with semi-colon (e.g. return a+b; | return 5; | return “Hi”; | …)

13) Condition_Operator: ( less than “<” | greater than “>” | is equal “=” | not equal “<>”)

14) Boolean_Operator: AND operator “&&” and OR operator “||”

15) If_Statement: “if” followed by Condition then reserved keyword “then” followed by set of Statements (i.e. any type of statement: write, read, …) then Else_If_Statment or Else_Statment or reserved keyword “end”

16) Repeat_Statement: starts with reserved keyword “repeat” followed by a set of Statements then reserved keyword “until” followed by Condition_Statement

## Code Sample

```
/*Sample program includes all rules*/

int sum(int a, int b)
{
	return a + b;

}
int main()
{
	int val, counter;
	read val;
	counter:=0;
	repeat
		val := val - 1;
		write "Iteration number [";
		write counter;
		write "] the value of x = ";
		write val;
		write endl;
		counter := counter+1;
	until val = 1
	write endl;
	string s := "number of Iterations = ";
	write s;
	counter:=counter-1;
	write counter;
	/* complicated equation */
	float z1 := 3*2*(2+1)/2-5.3;
	z1 := z1 + sum(1,y);
	if  z1 > 5 || z1 < counter && z1 = 1 then
		write z1;
	elseif z1 < 5 then
		z1 := 5;
	else
		z1 := counter;
	end
	return 0;
}
```

## Code Sample

```
/* Sample program in Tiny language - computes factorial*/
int main()
{
        int x;
        read x; /*input an integer*/
        if x > 0 then /*don't compute if x <= 0 */
                int fact := 1;
                repeat
                        fact := fact * x;
                        x := x – 1;
                until x = 0
                write fact; /*output factorial of x*/
        end
        return 0;
}
```