Pseudo-code

Matúš Kyseľ

Martin Orem

Lukáš Turčan

Pseudo-code

- The idea of this language is based on simple fact that all algorithms are already written in pseudo code.
- The pseudo code itself is not strictly defined yet.
- We decided to write a simple interpreter for the most common style.

Build && Requirements

- To build this interpreter just run bash script ./make.bash.
- For building are necessary: flex, bison and gcc.

Basic operations

- Pseudo-Code supports basic arithmetic operations as +-/ *<>!=.
- Each variable must be defined this way *A*
- Pseudo code currently supports only integer.
- Arrays can be defined similarly *A = 1,2,3*

Loops

- This language supports only one type of loops.
- A basic construction of for a loop:

```
for i from 1 to N do
    print A[i]
end for
```

Conditions

This language supports just basic if and else conditions.

```
if A > B then
    //DO SOMETHING
end if
```

Advanced condition

```
if A > B then
    //DO SOMETHING
else
    //DO SOMETHING ELSE
end if
```

Functions

Functions are defined just with special keyword *func* and every function must be ended with the *end func* statement.

```
func foo( A )
    foo = A + 1
end func
```

Simple bubble sort on array of integers

```
func bubblesort( )
   a = 4,3,2,1
    for i from 1 to len(a) do
        for j from 0 to len(a) -1 do
          if a[j] > a[j + 1] then
             swap( a[j], a[j + 1] )
          end if
        end if
    end for
end func
```

Thank you.