# Milestone3

Jhalak(210474) Mansi(210590) Prapti(210748)

17 April 2024

## 1 Installing the required files:

Implementation of Lexer and Parser is in Flex and Bison. So install the following if not previously installed-(in Ubuntu)

```
sudo apt-get update
sudo apt-get install flex
sudo apt-get install bison
```

## 2 Instructions for execution:

The command line options include-

1. -i or –input : give the file name from which input is to be read
2. -t or –tac : File name to which compiler outputs the 3AC
3. -s or –asm : File name to which compiler outputs the assembly code
4. -v or –verbose : Prints the derivation (parse tree) on the command line
5. -h or –help : Prints the manual page

For compilation-

```
$make
$./parser -i test.py -t test.3ac -s test.s>debug.txt
$gcc -c test.s -o test.o -no-pie; gcc test.o -o test -no-pie ;
$./test > result.txt
```

- On running the above commands, where input file is test.py, 3AC goes into test.3ac and assembly file goes into test.s.
- And the result obtained from assembly goes into result.txt

We have submitted 5 test cases. Few points regarding the test cases -

- Even after several changes, the lexer seems to give syntax error on encountering extra spaces or unnecessary newlines.
- If you face syntax error for the testcase where it's not supposed to be there, you are requested to remove the extra spaces or unnecessary newlines, if any.

No manual changes required to the generated assembly file or to run it successfully.

# 3 Language features supported:

- We implemented a statically typed python language.
- We assumed that all uses of variables for the first time will include the type and will be initialized hence if it's followed, error will be shown.
- In assembly, we supported for int, 1-D list, bool, strings, classes(single and multi level inheritance and constructors)
- All basic operators given below are supported for int and bool(wherever applied):
  Arithmetic operators: $+, -, , /, //, \%, *$

  Relational operators: $==, !=, >, <, >=, <=$

  Logical operators: $and, or, not$

  Bitwise operators: $\&, |, \wedge, \sim, <<, >>$

  Assignment operators: $=, +=, -=, =, /=, //=, \%=, *=, \&=, |=, \wedge=, <<=, >>=$
- Control flow via if-elif-else, for, while, break and continue.
- Supported the function len() for lists and strings.
  - Len works for lists and strings declared in the same scope.
- Supported Range() function with 1 or 2 arguments.
- Supported library function print() for int, bool, list access, basic operators on them and strings.
- Supported Recursion.
- Function arguments can be list, objects, list access, int, bool, string.
- Methods and method calls, including both static and non-static methods
- We assumed that main() function will always be defined.

## 4 Modifications of 3AC from Milestone2 to Milestone3:

No modifications

## 5 Required features- Unsupported

- Static polymorphism via method overloading isn't supported.
- Relational operators on strings
- Multiple level inheritance isn't supported but single level is supported.

## 6 Effort Sheet:

- Jhalak (210474) - 33.33%
- Mansi (210590) - 33.33%
- Prapti (210748) - 33.33%