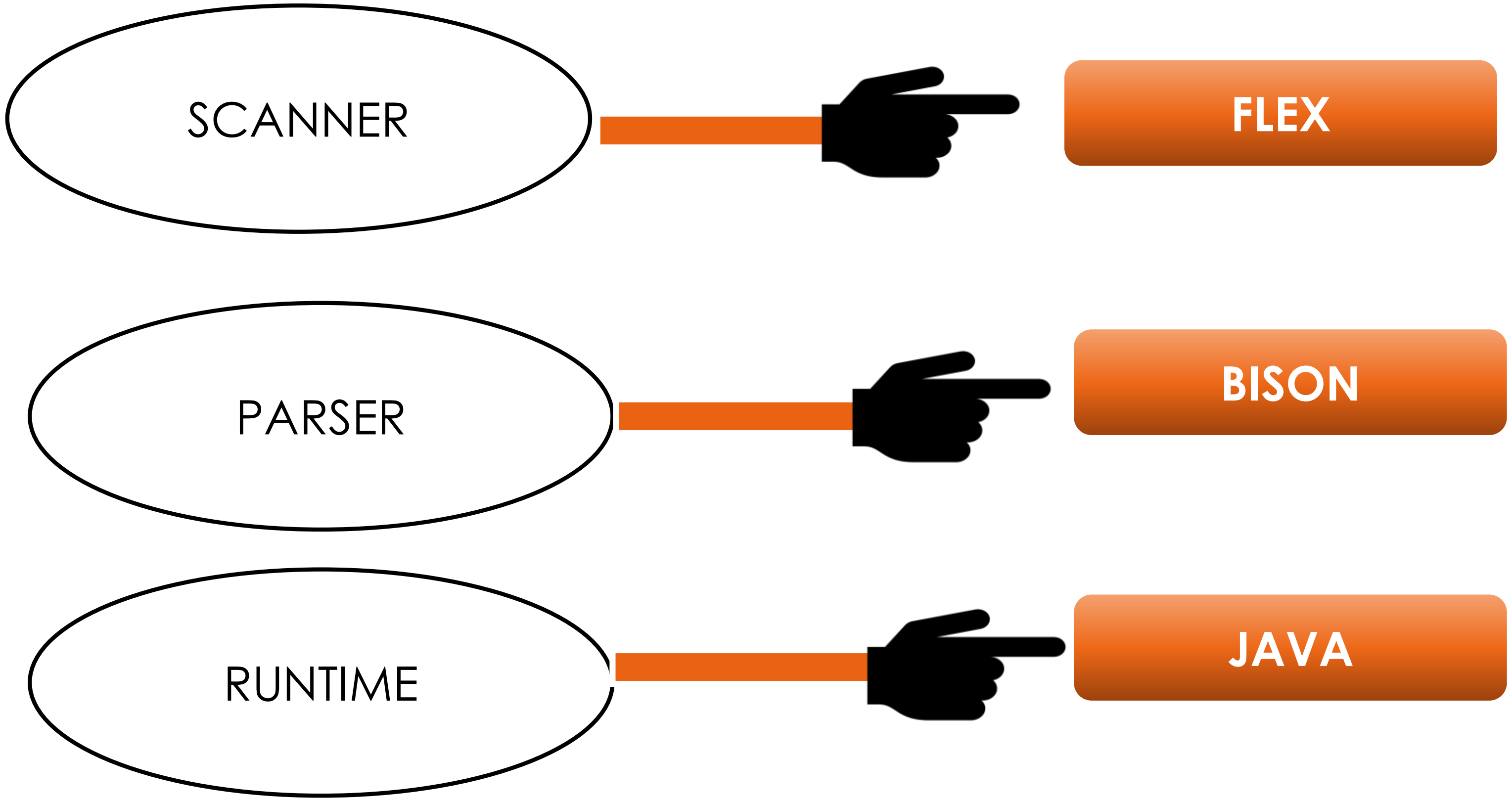
COMPILER FOR PROGRAMMING LANGUAGE- ORANGE

TEAM – 23

ABHISHEK NAGARAJ
DEEPAK PARAMESHWARAN
SNEHA VIDHYASHEKAR

UNDER THE GUIDANCE OF
PROFESSOR AJAY BANSAL

TOOLS USED

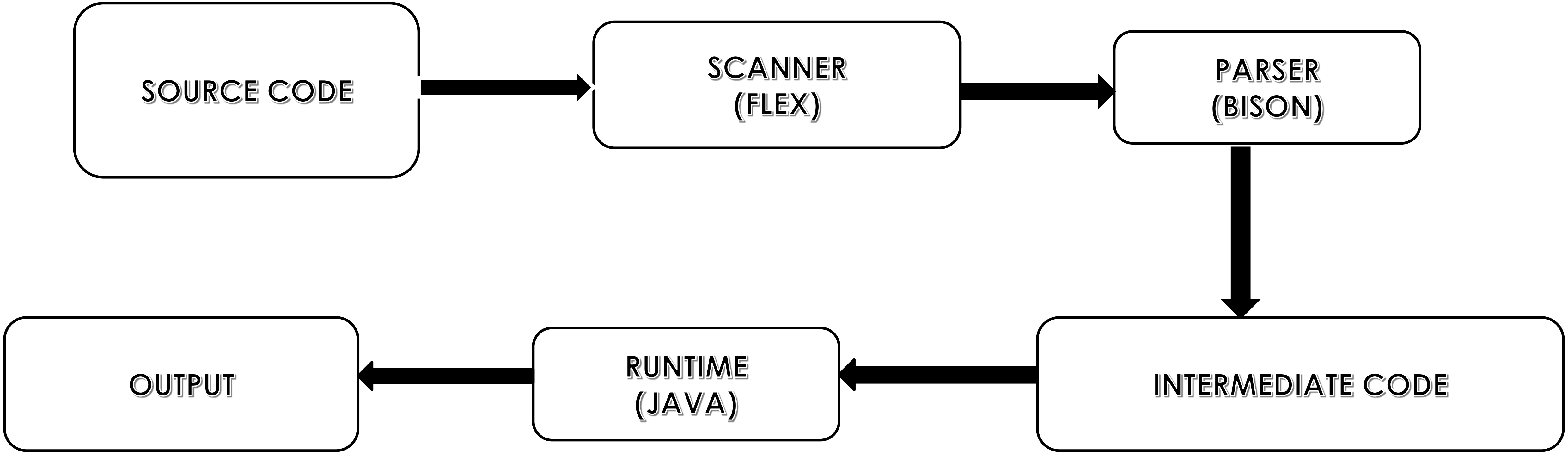


SCANNER ☞ FLEX

PARSER ☞ BISON

RUNTIME ☞ JAVA

COMPILER – DESIGN FLOW



OPERATORS	DESCRIPTION
“+”	Addition
“-”	Subtraction
“*”	Multiplication
“/”	Division
“AND”	AND
“OR”	OR
“NOT”	NOT
“->”	Assignment operator
“~”	delimiter
“->->”	Comparison operator
“!->”	Not Equal to
“->>”	Greater than equal to
“->“	Leasser than equal to
“>”	Greater than
“<”	Lesser than
“YES”	True
“NO”	False
“LP“	Left parentheses

KEYWORDS

num	Number(int/float)
read	Input
write	output
nl	New line
is	if
isnot	else
loop	While loop

STEPS TO RUN "ORANGE" PROGRAMMING LANGUAGE

Command 1

```
bison –dy ORJ.y
```

- To generate the parser which generates y.tab.c and y.tab.h files

Command 2

```
flex ORJ.l
```

- This is the scanner which in turn generates lex.yy.c

Command 3

```
gcc –o xyz.orj lex.yy.c y.tab.c -lm
```

- This generates xyz.orj file which helps to generate an intermediate code for the program

Command 4

```
xyz.orj test.orj
```

- To generate an intermediate code by making use of xyz.orj file which has an extension ".int"

Command 5

```
javac Runtime.java
```

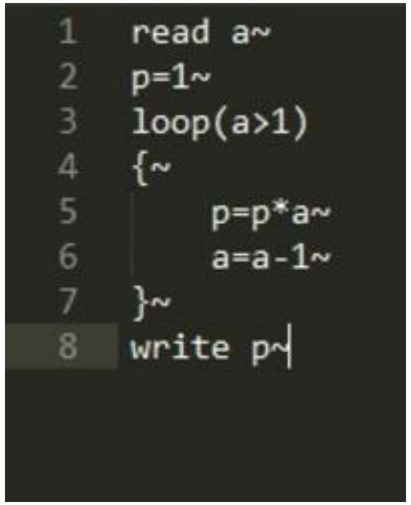
- To run the java file which generates a class file to execute the program.

Command 6

```
java Runtime test.orj.int
```

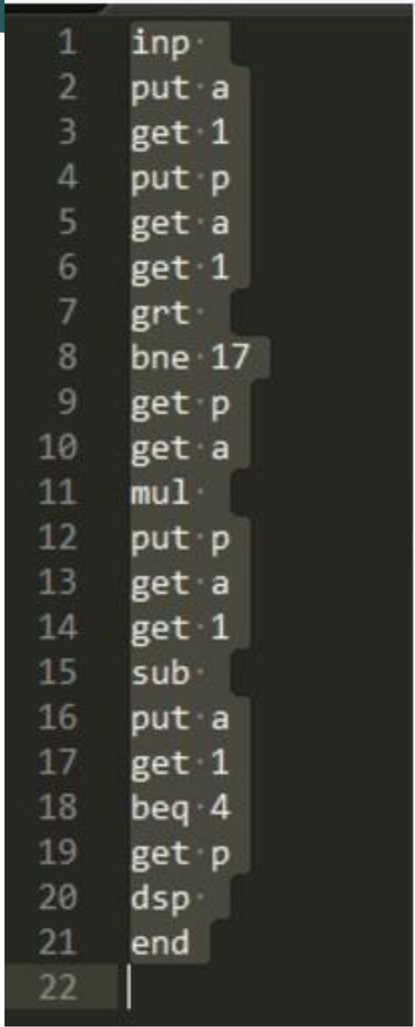
- To run/execute the program(Intermediate code) in the java environment and give the output.

SOURCE CODE



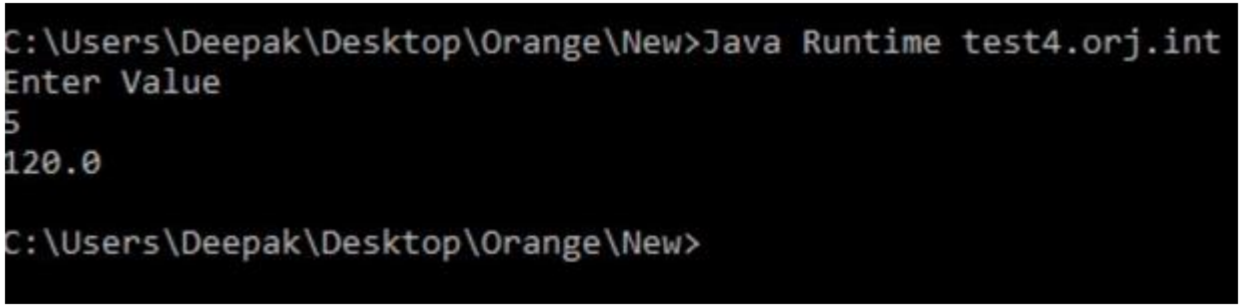
```
read a~
p=1~
loop(a>1)
{~
	p=p*a~
	a=a-1~
}~
write p~
```

INTERMEDIATE CODE



```
 1  inp
 2  put a
 3  get 1
 4  put p
 5  get a
 6  get 1
 7  grt
 8  bne 17
 9  get p
10  get a
11  mul
12  put p
13  get a
14  get 1
15  sub
16  put a
17  get 1
18  beq 4
19  get p
20  dsp
21  end
22
```

OUTPUT



```
C:\Users\Deepak\Desktop\Orange\New>Java Runtime test4.orj.int
Enter Value
5
120.0

C:\Users\Deepak\Desktop\Orange\New>
```

THANK YOU!