# Cycle 2

Malavika R Vikraman

CS7B – Roll No :31

## **Question 6**

Write a YACC program to check the syntax of FOR statement in C

Program :

Lex :

```
%{
    #include "y.tab.h"
%}

%%
for                     return FOR;
\< |
\<= |
== |
\>= |
\> |
!=                      return RELOP;
\+\+                    return INC;
--                      return DEC;
\|\| |
&&                      return LOGOP;
[a-zA-Z_][a-zA-Z_0-9]*          return ID;
[0-9]+                  return NUM;
[\t\n ]                 ;
.                       return yytext[0];
%%

int yywrap() {}
```

Yacc :

```
%{
    #include<stdio.h>
    #include<stdlib.h>
    extern FILE *yyin;
%}

%token FOR RELOP LOGOP ID NUM INC DEC;
%right '='
```

```
%left LOGOP
%left RELOP
%left '+' '-'
%left '*' '/'
%left INC DEC

%%
start:        st                                    { printf("Input Accepted!\n");
exit(0); }
st:           FOR '(' loop_expr ';' condition ';' loop_expr ')' block ;
loop_expr:        E | ;
block:        '{' code '}' | statement ;
code:         statement code | statement ;
statement:    E ';' | st ;

E:            ID '=' E
              |E '+' E
              | E '-' E
              | E '*' E
              | E '/' E
              | E RELOP E
              | E LOGOP E
              |ID INC
              | ID DEC
              | ID
              | NUM
              ;

condition:    E RELOP E
              | E LOGOP E
              |;

%%

int yyerror(char* s) {
   fprintf(stderr, "%s\n", s);
}

int main(int argc, char* argv[]) {
   FILE *fp;
   if(argc>1) {
      fp=fopen(argv[1],"r");
      if(!fp) {
         perror(fp);
         printf("%s not found", argv[1]);
         exit(0);
      }
```

```
      yyin=fp;
   }
   printf("Checking the validity of for loop...\n");
   yyparse();
}
```

**Input:**

```
1    for (i=0; i < 100 ; i++) {
2        temp = i + 10;
3        a = b + c;
4    }
```

**Output :**

```
C:\Users\malav\OneDrive\Documents\CDLab\Cycle2\Pg7>a input.c
Checking the validity of for loop...
Input Accepted!
```