Practical 3: Write a LEX / YACC program for Symbol Table Generation

Gahan Saraiya (18MCEC10)

18mcec10@nirmauni.ac.in

I. AIM

This is the symbol table implementation in yacc using regular expressions, it checks for redeclaration of variables and multiple declarations of variables

I. Implementation

```
%{
#include
#include "y.tab.h"
#include
int fl=0,i=0,type[100],j=0,error_flag=0;
char symbol[100][100],temp[100];
%}
%token INT FLOAT C DOUBLE CHAR ID NL SE O
%%
START:S1 NL {return;}
;
S1:S NL S1
|S NL
;
S:INT L1 E
|FLOAT L2 E
|DOUBLE L3 E
|CHAR L4 E
|INT L1 E S
|FLOAT L2 E S
|DOUBLE L3 E S
|CHAR L4 E S
|O
;
L1:L1 C ID {strcpy(temp,(char *)$3);insert(0);}
|ID {strcpy(temp,(char *)$1);insert(0);}
;
```

```
L2:L2 C ID {strcpy(temp,(char *)$3);insert(1);}
|ID {strcpy(temp,(char *)$1);insert(1);}
;
L3:L3 C ID {strcpy(temp,(char *)$3);insert(2);}
|ID {strcpy(temp,(char *)$1);insert(2);}
;
L4:L4 C ID {strcpy(temp,(char *)$3);insert(3);}
|ID {strcpy(temp,(char *)$1);insert(3);}
;
E:SE
;
%%
main()
{
yyparse();
if(error_flag==0)
for(j=0;j
{
if(type[j]==0)
printf(" INT - ");
if(type[j]==1)
printf(" FLOAT - ");
if(type[j]==2)
printf(" DOUBLE - ");
if(type[j]==3)
printf(" CHAR - ");
printf(" %s\n",symbol[j]);
}
}
void yyerror()
{ printf("SYNTAX ERROR\n");
error_flag=1;
}
void insert(int type1)
{
fl=0;
for(j=0;j
if(strcmp(temp,symbol[j])==0)
{
if(type[i]==type1)
printf("REDECLARATION OF %s\n",temp);
else
{
printf("MULTIPLE DECLARATION OF %s\n",temp);
error_flag=1;
}
fl=1;
}
if(fl==0)
{
strcpy(symbol[i],temp);
type[i]=type1;
i++;
}
}
```

II. Output

```
char tt
SYNTAX ERROR
float gg;
char dd,ff;
int ll;
FLOAT - gg
CHAR - dd
CHAR - ff
INT - ll
oijfp
int xx;
float xx;
MULTIPLE DECLARATION OF xx
```