GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

Experiment No: 11

Symbol Table

Aim: Write a YACC program to implement symbol table for given input.

Theory:

Symbol table is an important data structure created and maintained by compilers in order to store information about the occurrence of various entities such as variable names, function names, objects, classes, interfaces, etc. Symbol table is used by both the analysis and the synthesis parts of a compiler.

A symbol table may serve the following purposes depending upon the language in hand:

- To store the names of all entities in a structured form at one place.
- To verify if a variable has been declared.
- To implement type checking, by verifying assignments and expressions in the source code are semantically correct.
- To determine the scope of a name (scope resolution).

Lex Program:

```
%{
#include<stdio.h>
#include "y.tab.h"
int yyval;
int c;
%}

%%
int return INT;
float return FLOAT;
char return CHAR;
"," return COMMA;
";" return SC;
[a-z]+ {c = yytext[0]; yylval = c; return ID;}
\n return NL;
%%
```

Yacc Program:

```
%{
#include<stdio.h>
#include<string.h>
char* temp[50];
%}
```

GOA COLLEGE OF ENGINEERING

"Bhausaheb Bandodkar Technical Education Complex"

```
%token ID INT FLOAT CHAR SC NL COMMA

%%
PROGRAM:
      PROGRAM START NL {printf("Valid Decalaration Statement.\n");}
      |
      ;
START:
      INT INTVAR
      |FLOAT FLOATVAR
      |CHAR CHARVAR
      ;
INTVAR:
      ID SC { sprintf(temp,"\n%c\tint\t%d",$1,sizeof(int));Gen(temp); }
      | ID COMMA INTVAR { sprintf(temp,"\n%c\tint\t%d",$1,sizeof(int));Gen(temp); }
      ;
FLOATVAR:
      ID SC { sprintf(temp,"\n%c\tfloat\t%d",$1,sizeof(float));Gen(temp); }
      | ID COMMA FLOATVAR { sprintf(temp,"\n%c\tfloat\t%d",$1,sizeof(float));Gen(temp);}
      ;
CHARVAR:
      ID SC { sprintf(temp,"\n%c\tchar\t%d",$1,sizeof(char));Gen(temp); }
      | ID COMMA CHARVAR { sprintf(temp,"\n%c\tchar\t%d",$1,sizeof(char));Gen(temp);}
      ;
%%

void yyerror()
{
printf("Error\n");
}

int yywrap()
{
return 1;
}

void Gen(char *val)
{
FILE *fo ;
fo=fopen("output.txt","a");
fputs(val,fo);
fclose(fo);
}
```

```
main()
{
yyparse();
}
```

Conclusion:

The yacc program to implement symbol table has been successfully executed.