Practical - 6

AIM: Write a program to validate the variable declaration statement using Lex and YACC.

Program:

1) Yacc File

```
%{
#include<stdio.h>
%}
%token ID DT SC COMMA

%%
start:DT varlist SC {printf("valid");}
|
varlist:varlist COMMA ID|ID;
%%

void yyerror(const char *str)
  {printf("error");}

int yywrap(){
return 0;
}

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

2) Lex File

```
%{
#include<stdio.h>
#include "y.tab.h"
%}
%%
int|float|char        { return DT; }
","                   { return COMMA; }
";"                   { return SC; }
[a-zA-Z]+[a-zA-Z0-9]*  { return ID; }
[ \t\n]+              { /* Ignore whitespace */ }
.                     { printf("Invalid character\n"); }
%%
```

Output:

```
[21012021001@linuxserv ~]$ nano cdpr6.y
[21012021001@linuxserv ~]$ yacc -d cdpr6.yacc
yacc: f - cannot open "cdpr6.yacc"
[21012021001@linuxserv ~]$ yacc -d cdpr6.y
[21012021001@linuxserv ~]$
[21012021001@linuxserv ~]$ nano cdpr6.y
[21012021001@linuxserv ~]$ yacc -d cdpr6.y
[21012021001@linuxserv ~]$ nano cdpr6.lex
[21012021001@linuxserv ~]$ lex cdpr6.lex
[21012021001@linuxserv ~]$ cc lex.yy.c y.tab.c -o op
[21012021001@linuxserv ~]$ ./op
int r
int r;
error[21012021001@linuxserv ~]$ cc lex.yy.c y.tab.c
[21012021001@linuxserv ~]$ ./a.out
int r;
valid
int a=21;
error[21012021001@linuxserv ~]$ ./a.out
int a,r;
valid
int a
error[21012021001@linuxserv ~]$
```