Exp 7 : YACC – VALID IDENTIFIER

LEX

1. Start
2. Rules Section
    1. [0-9]+ : matches one or more digits and returns the token type digit
    2. [a-zA-Z]+ : matches one oe more alphabets and returns the token type letter
    3. [\t\n] : skips whitespaces
    4. . : matches any other character and returns it as a single character token
3. yywrap() indicate the end of input
4. Stop

YACC

1. Start
2. Token Declarations (%token) : digit & letter
3. Grammar Rules Section
    - The first character would be a letter and it can be followed by any no of letters and digits
    - S : letter A
    - A : letter A | digit A
4. yyerror() to handle
5. main() prompts the user to enter an identifier and calls yyparse() to parse it
6. Stop

Valid Identifier (Lex)

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

%%
[0-9]+ { return digit; }
[a-zA-Z]+ { return letter; }
[ \t\n] {;}
. { return yytext[0]; }
%%

int yywrap(){
   return 1;
}
```

Valid Identifier (YACC)

```
%{
   #include<stdio.h>
   int flag = 0;
%}
%token digit letter

%%
S : letter A;
A : letter A | digit A | ;
%%

int yyerror(){
   flag = 1;
   printf("invalid identifier\n");
   return 1;
}
int main(){
   printf("Enter the identifier : ");
   yyparse();
   if(flag == 0){
      printf("valid identifier\n");
   }
   return 0;
}
```

Note : if output is not printed then press ctrl+D