Aim: Program to implement Desk calculator with error recovery.

Lex code:

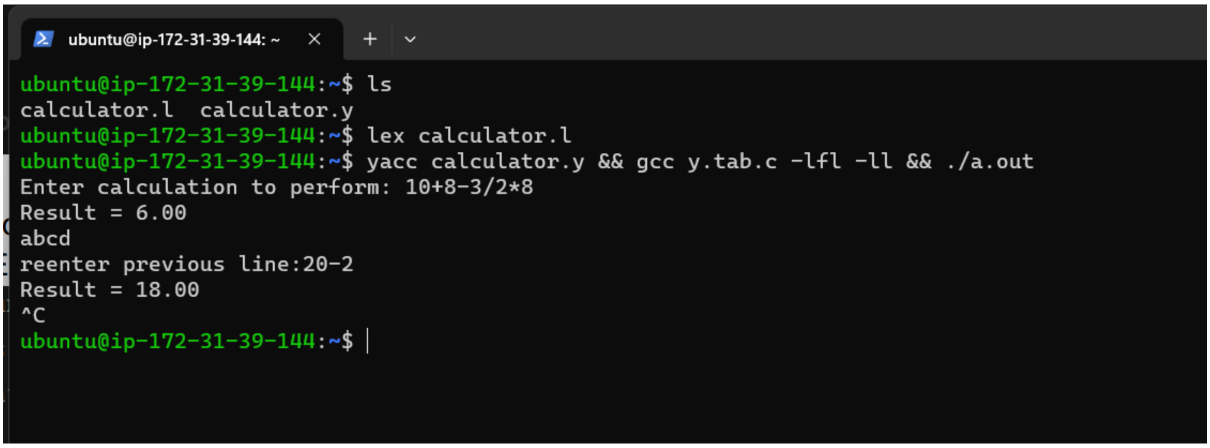
```
num [0-9]+\.?|[0-9]*\.[0-9]+
%%
{num} { yylval = (double)atoi(yytext); return num; }
[ ] {}
\n|. { return yytext[0]; }
%%
```

Yacc Code:

```
%{
#include <stdio.h>
#include <stdlib.h>
#define YYSTYPE double
int yylex(void);
void yyerror(char const* s);
void push();
%}
%token num
%left '+' '-'
%left '*' '/'
%right UMINUS
%%
S : S E '\n' {printf("Result = %.2f\n", $2);}
| S '\n'
```

```
|
| error '\n' { yyerrok; }
;
E : E'+'E {$$ = $1 + $3;}
| E'-'E {$$ = $1 - $3;}
| E'*'E {$$ = $1 * $3;}
| E'/'E {$$ = $1 / $3;}
| '('E')' { $$ = $2; }
| '-' E %prec UMINUS { $$ = -$2; }
| num {$$ = $1;}
;
%%
#include"lex.yy.c"
void yyerror (char const *s) {
printf("reenter previous line:");
}
int main()
{
printf("Enter calculation to perform: ");
yyparse();
return 0;
}
```

Output:



```
ubuntu@ip-172-31-39-144:~$ ls
calculator.l  calculator.y
ubuntu@ip-172-31-39-144:~$ lex calculator.l
ubuntu@ip-172-31-39-144:~$ yacc calculator.y && gcc y.tab.c -lfl -ll && ./a.out
Enter calculation to perform: 10+8-3/2*8
Result = 6.00
abcd
reenter previous line:20-2
Result = 18.00
^C
ubuntu@ip-172-31-39-144:~$ 
```