pfcalc.y

```
%{
    #include <stdio.h>
    #include <stdlib.h>

    int yylex(void);
    int yyerror(char *);
%}

%token NUMBER EOL

%%
input: /* empty */
    | input expr EOL { printf("= %d\n", $2); }
    ;

expr: NUMBER {$$ = $1}
    | expr expr '+' {$$ = $1 + $2}
    | expr expr '-' {$$ = $1 - $2}
    | expr expr '*' {$$ = $1 * $2}
    | expr expr '/' {$$ = $1 / $2}
;
%%

int main() {
    yyparse();
    return 0;
}

int yyerror(char *s) {
    fprintf(stderr, "![E] %s\n", s);
    return 0;
}
```

pfcalc.l

```

%option noyywrap
%{
    #include <stdio.h>

    #include "pfcalc.tab.h"
%}

%%

[ \t]+ { /* Ignore Whitespaces */}

[0-9]+ { yylval = atoi(yytext); return NUMBER; }

[+\-*/] { return *yytext; }

\n { return EOL; }

%%
```

Output

```
PS C:\DevParapalli\Projects\RTMNU-SEM-6> cd '.\CD\Practical 07\'
PS C:\DevParapalli\Projects\RTMNU-SEM-6\CD\Practical 07> flex pfcalc.l && bison -d pfcalc.y &&
  gcc lex.yy.c pfcalc.tab.c -o pfcalc.exe && ./pfcalc.exe
32 1 + 45 + 3 *
= 234
```