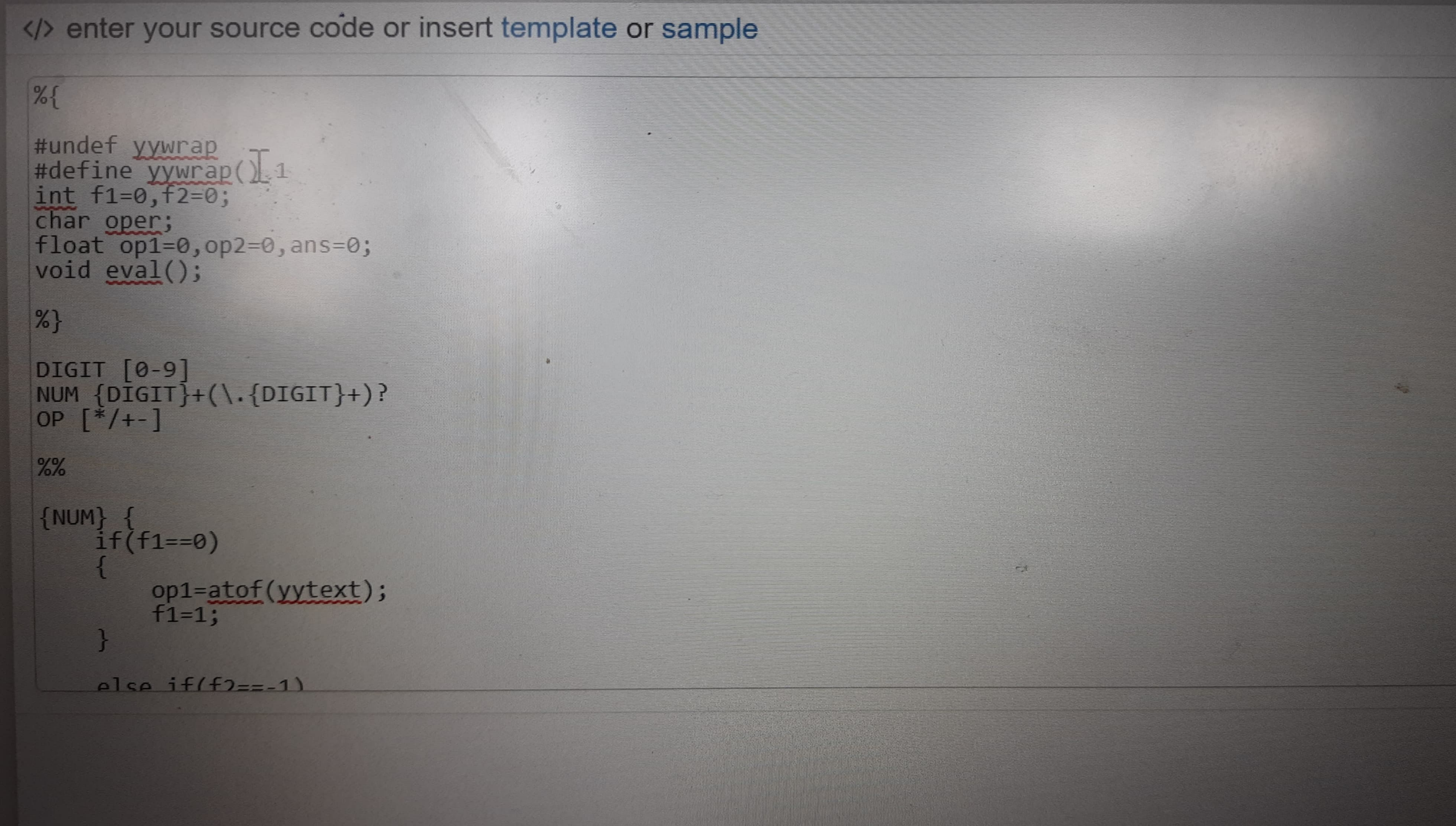
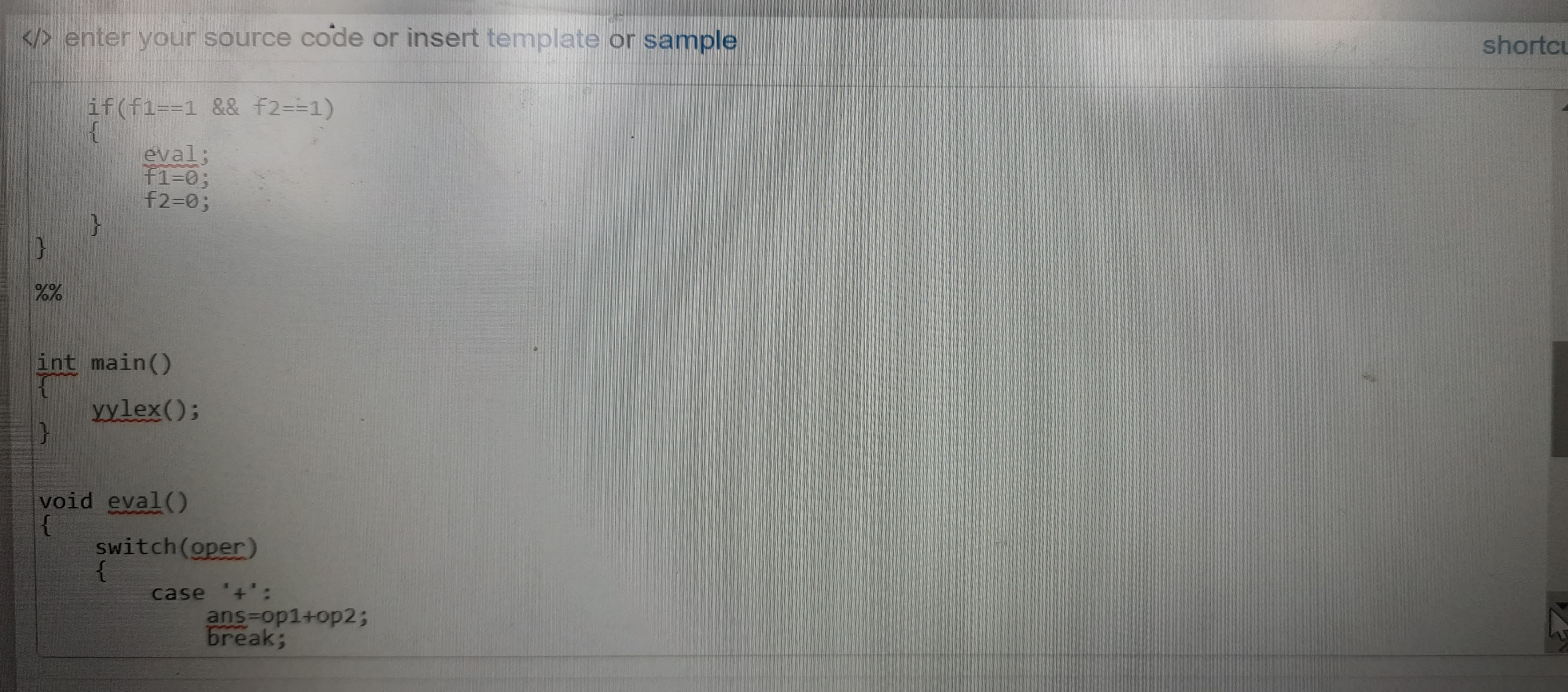


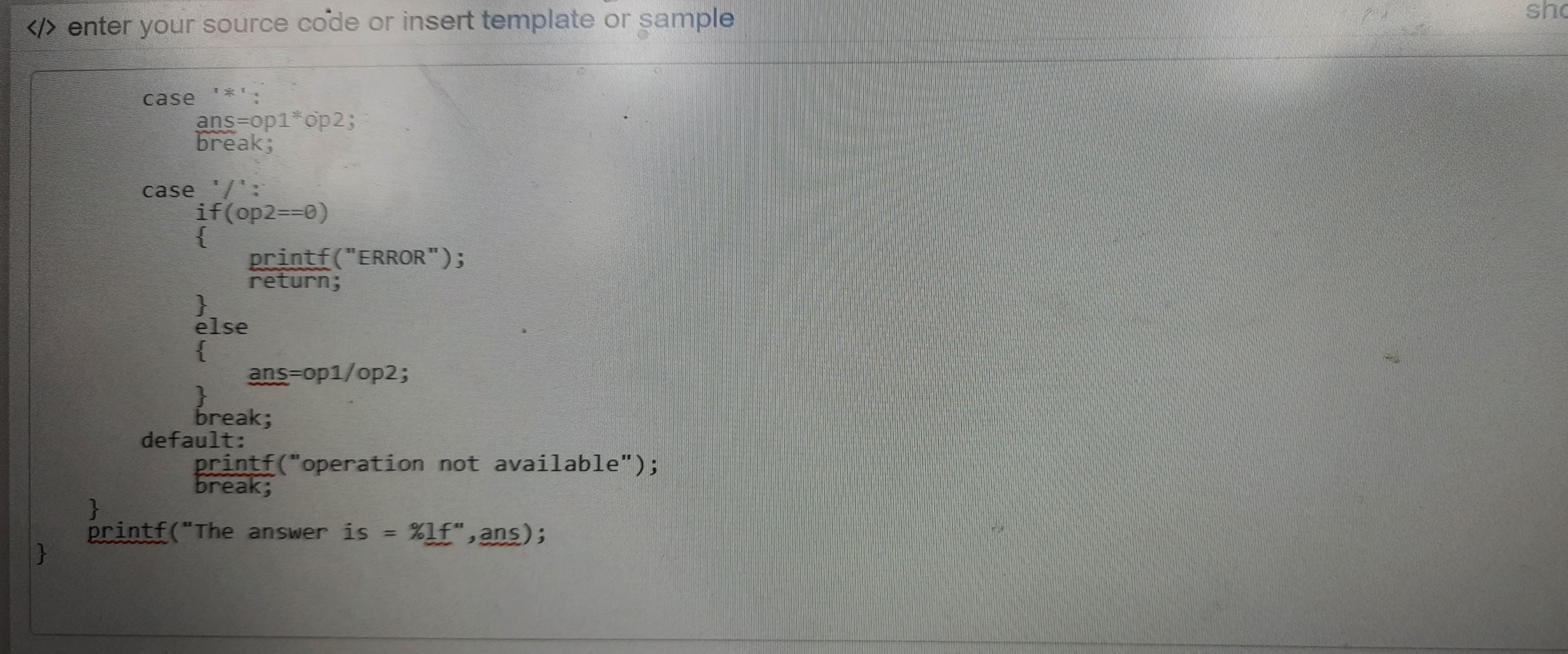
</> enter your source code or insert template or sample

```
%{

#undef yywrap
#define yywrap() 1
int f1=0,f2=0;
char oper;
float op1=0,op2=0,ans=0;
void eval();

%}

DIGIT [0-9]
NUM {DIGIT}+(\.{DIGIT}+)?
OP [*/+-]

%%

{NUM} {
        if(f1==0)
        {
                op1=atof(yytext);
                f1=1;
        }

        else if(f2==-1)
```

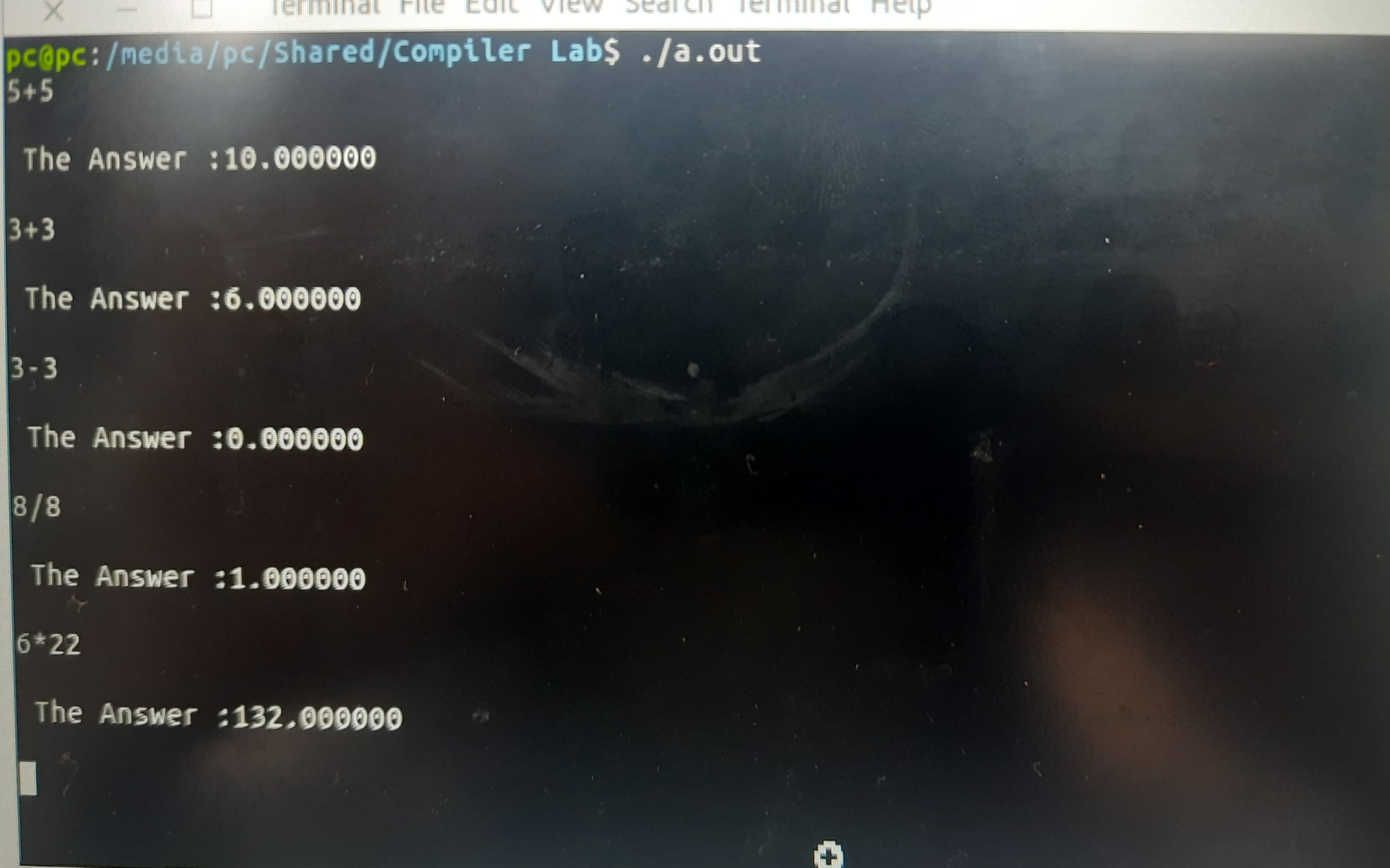
```
    else if(f2==-1)
    {
        op2=atof(yytext);
        f2=1;
    }

    if((f1==1) && (f2==1))
    {
        eval();
        f1=0;
        f2=0;

    }
}
{OP} {

    oper=(char) *yytext;
    f2=-1;
}
[\n] {

    if(f1==1 && f2==1)
```


&lt;/&gt; enter your source code or insert template or sample
shortcu

```
    if(f1==1 && f2==1)
    {
        eval;
        f1=0;
        f2=0;
    }
}

%%


int main()
{
    yylex();
}


void eval()
{
    switch(oper)
    {
        case '+':
            ans=op1+op2;
            break;
```


&lt;/&gt; enter your source code or insert template or sample
sho

```
        case '*':
            ans=op1*op2;
            break;

        case '/':
            if(op2==0)
            {
                printf("ERROR");
                return;
            }
            else
            {
                ans=op1/op2;
            }
            break;
        default:
            printf("operation not available");
            break;
    }
    printf("The answer is = %lf",ans);
}
```



```
pc@pc:/media/pc/Shared/Compiler Lab$ ./a.out
5+5

 The Answer :10.000000

3+3

 The Answer :6.000000

3-3

 The Answer :0.000000

8/8

 The Answer :1.000000

6*22

 The Answer :132.000000

```