This program is a simple calculator that reads its input from a character buffer. If integers are read, they are pushed on a stack. If one of the operators (+ - \* ⁄) is read, the top two elements are popped off the stack, the operation is performed on them, and the result is pushed on the stack. The = operator writes out the value of the top element of the stack to a buffer

.

BUGS: CALC.c CORRECTIONS MADE

Not present #include "PUSHPOP.c"

Not present #include "READTOKN.c"

include "calc.h" remove

Not present printf("%s", buf\_out);

PUSHPOP.c

#include "calc.h #include "CALC.h"

ptr- >i    = num; Ptr->i = *num*;

ptr–>next = stk–>top; ptr->next = *stk*->top;

stk–>top  = ptr; *stk*->top = ptr;

ptr      = stk–>top; ptr=*stk*->top;

num   = ptr–>i; num = ptr->i;

stk–>top = ptr–>next; *stk*->top=ptr->next;

for (c = nextchar(); for (c = nextchar();

READTKN.c

isspace(c); !isspace(c);

c = nextchar()); c = nextchar()){}

⁄\*\*⁄ /\*\*/ or //

*'⁄'* *‘/’*

Remove #include "calc.h"

CALC.h

typedef *enum* toks typedef *enum ,*

extern *void*  push(IntStack\*,*int*); extern *void* push(IntStack\**stk*,*int* *num*);

extern *int* pop(IntStack \*); extern *int* pop(IntStack \**stk*);

Steps we did:

Step1: *Read the entire code and understood the logic*

Step2: *Checked the code for syntactical errors*

Step3: *Checked the functions individually by inputs to attain the mentioned output and made corrections*

Step4: *Checked the header file for errors and corrected them*

Step5: Compilation and execution of program