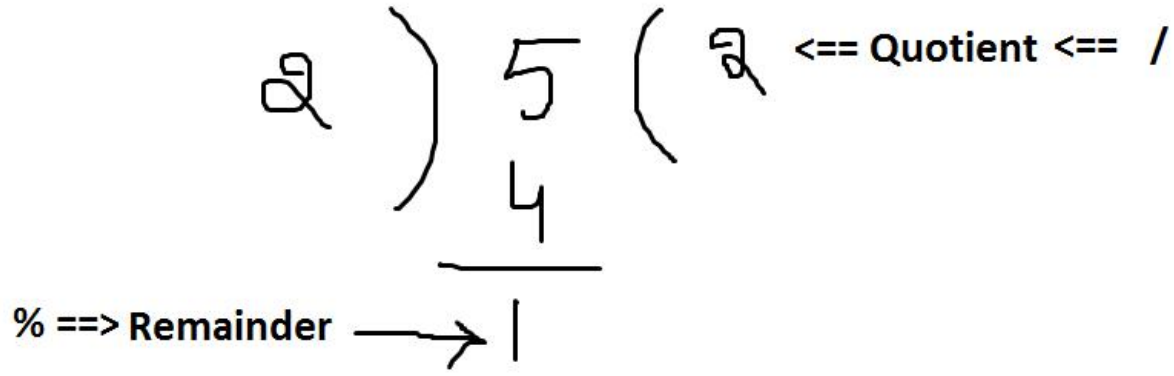
Arithmetic operators [+, -, *, %, /]: They are used to perform mathematical operations.

Eg: a+b, a-b, a*b,……

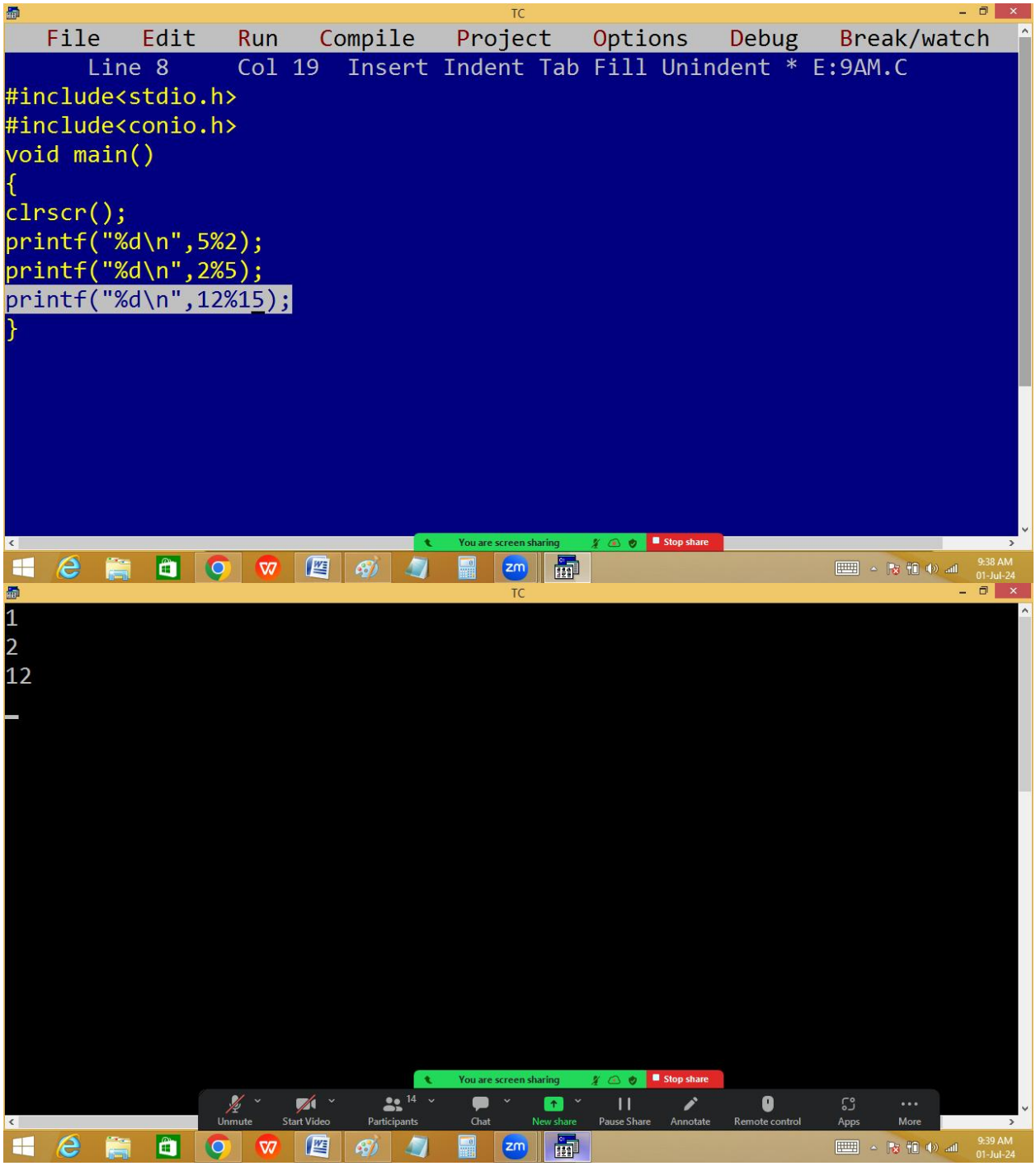
% - Modules [Remainder]:

5%2=1



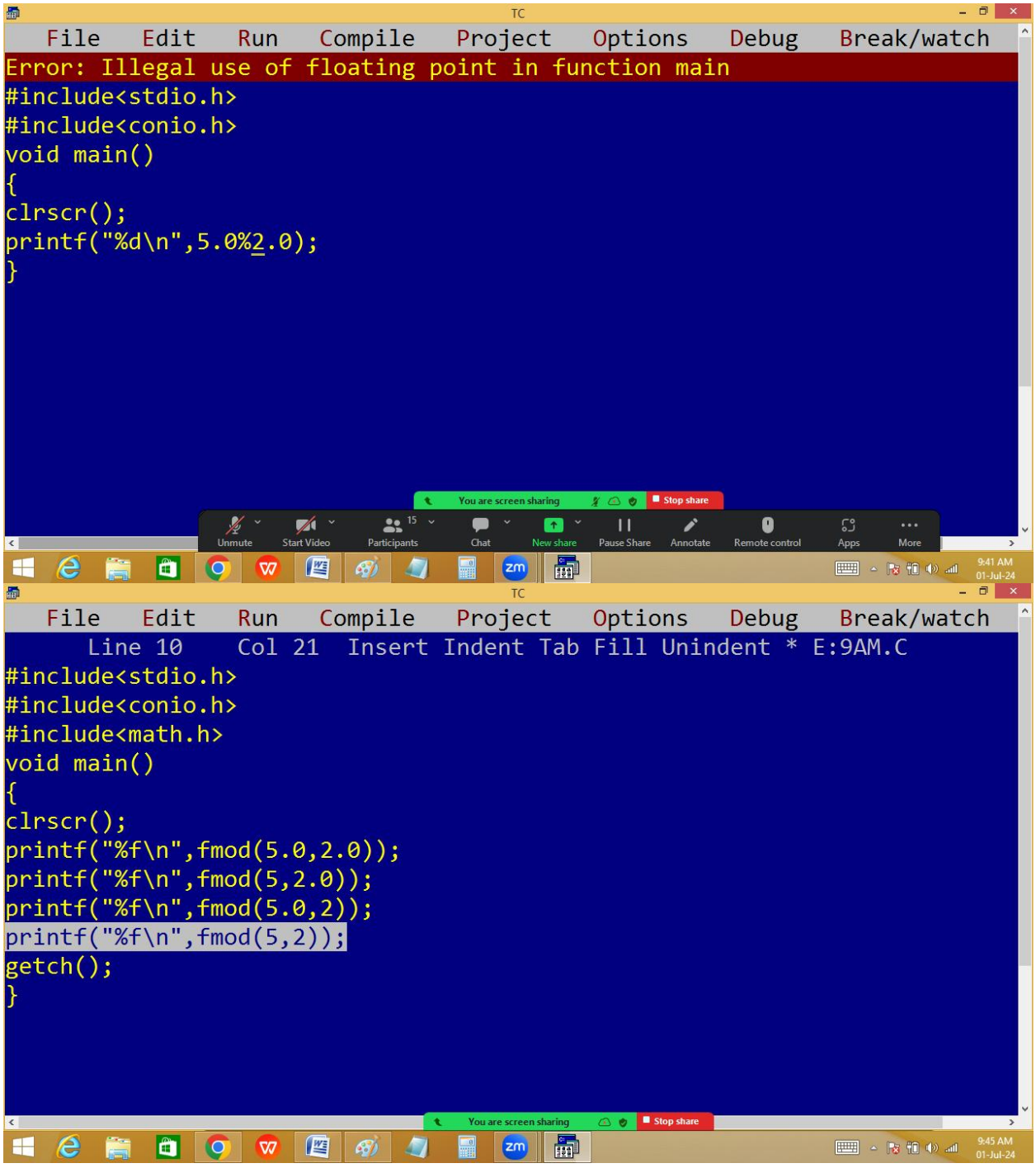
2%5=2

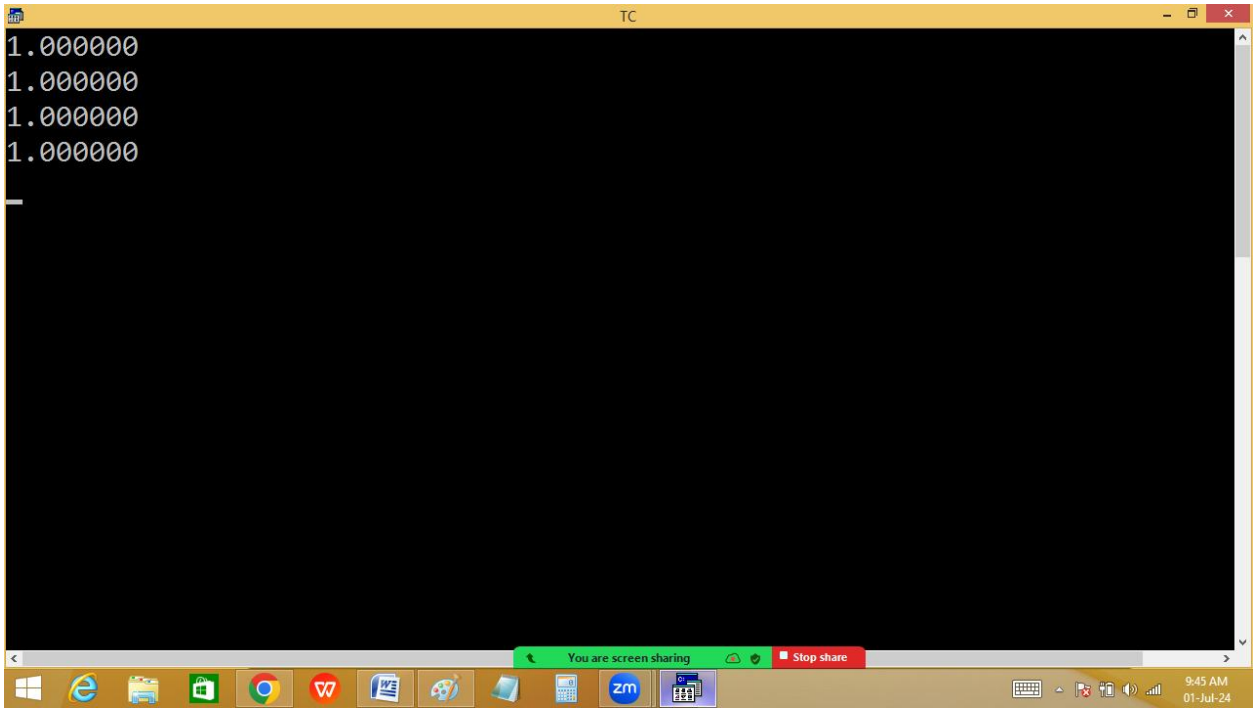
Note: If the divisor is bigger than dividend then the dividend is the answer.


TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 19 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n",5%2);
printf("%d\n",2%5);
printf("%d\n",12%15);
}
You are screen sharing
Stop share
TC
1
2
12
Unmute
Start Video
Participants
Chat
New share
Pause Share
Annotate
Remote control
Apps
More

5.0 % 2.0 = Error

Note: In C & C++ we can't perform floating modules with **%** operator. For this we have to use fmod() available in <math.h>


TC
File Edit Run Compile Project Options Debug Break/watch
Error: Illegal use of floating point in function main
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n",5.0%2.0);
}
You are screen sharing
Stop share
Unmute
Start Video
Participants
Chat
New share
Pause Share
Annotate
Remote control
Apps
More
9:41 AM
01-Jul-24
TC
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 21 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
printf("%f\n",fmod(5.0,2.0));
printf("%f\n",fmod(5,2.0));
printf("%f\n",fmod(5.0,2));
printf("%f\n",fmod(5,2));
getch();
}
You are screen sharing
Stop share
9:45 AM
01-Jul-24



291%10=1

34%10=4

3%10=3

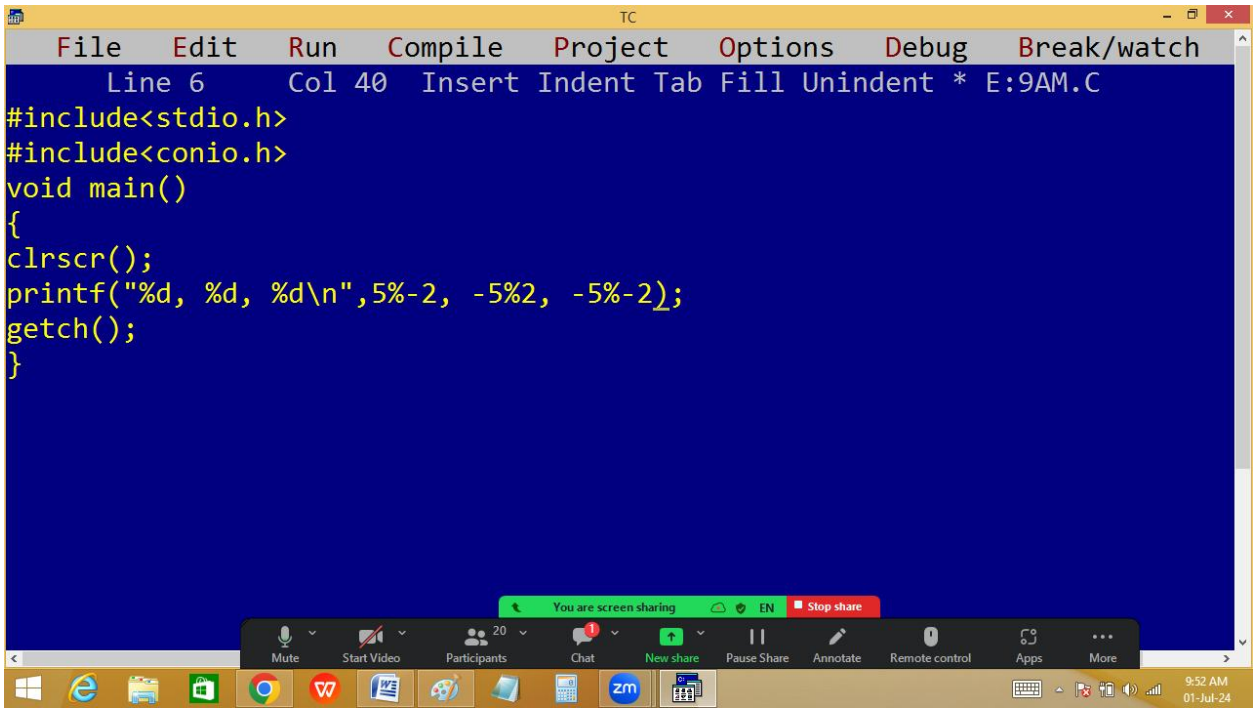
Note: Any no%10 gives the last digit.

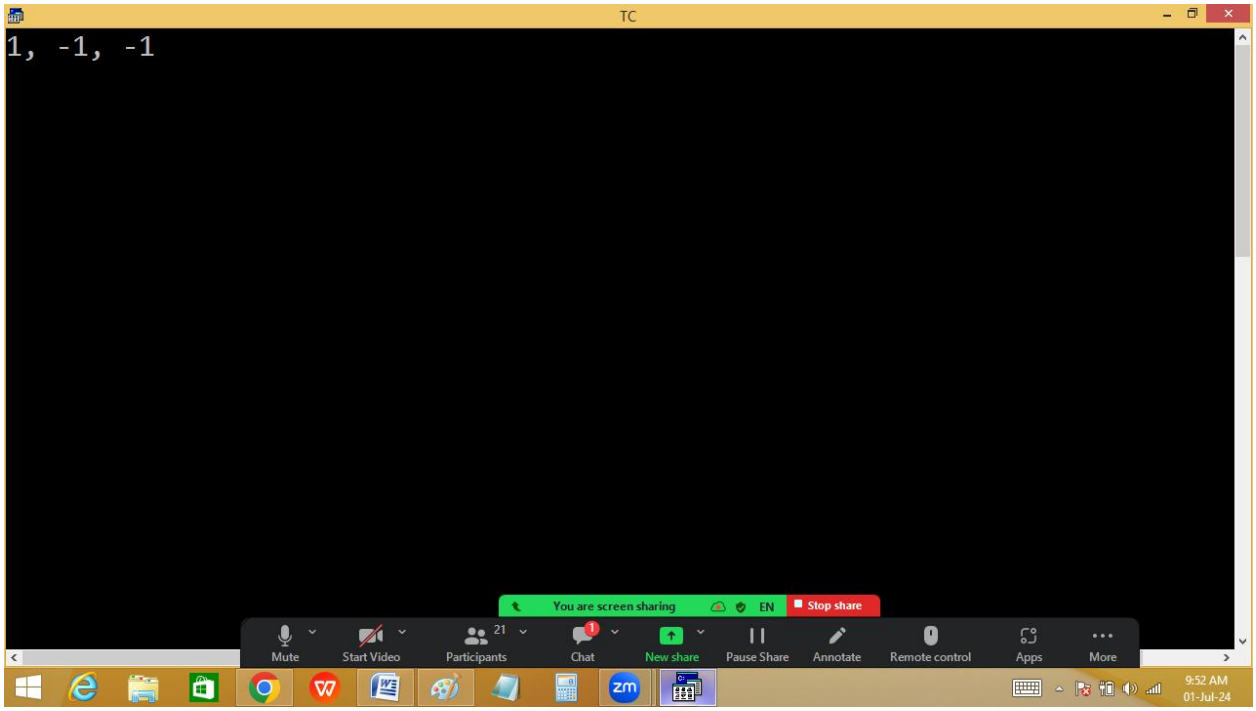
5%-2= 1

-5%2= -1

-5%-2= -1

Note: If the numerator is negative then result also negative.





/ - division [Quotient]:

5/2=2 [int/int=int]

5.0/2=2.500000

5/2.0=2.500000

5.0/2.0=2.500000

(float)5/2=2.500000 [explicit type casting]

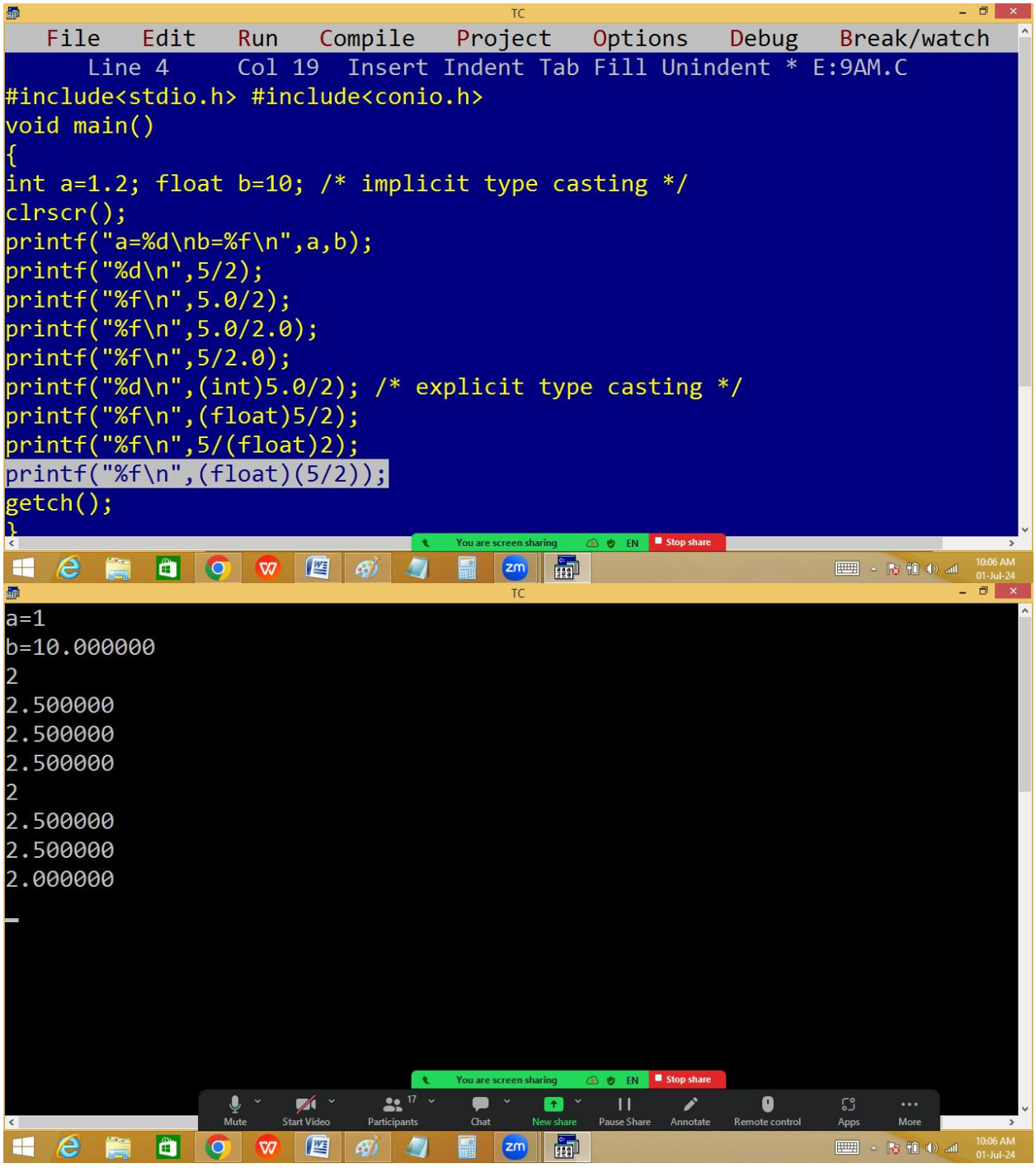
(int)5.0/2= [explicit type casting]

Int a=1.2; /* implicit type casting */ ➔ a=1

Float b=10; /* implicit type casting */ ➔ b=10.000000

(float)(5/2)=2.000000

Note: In C when both operands are int then result also int. if anyone or both are floats then result also float.

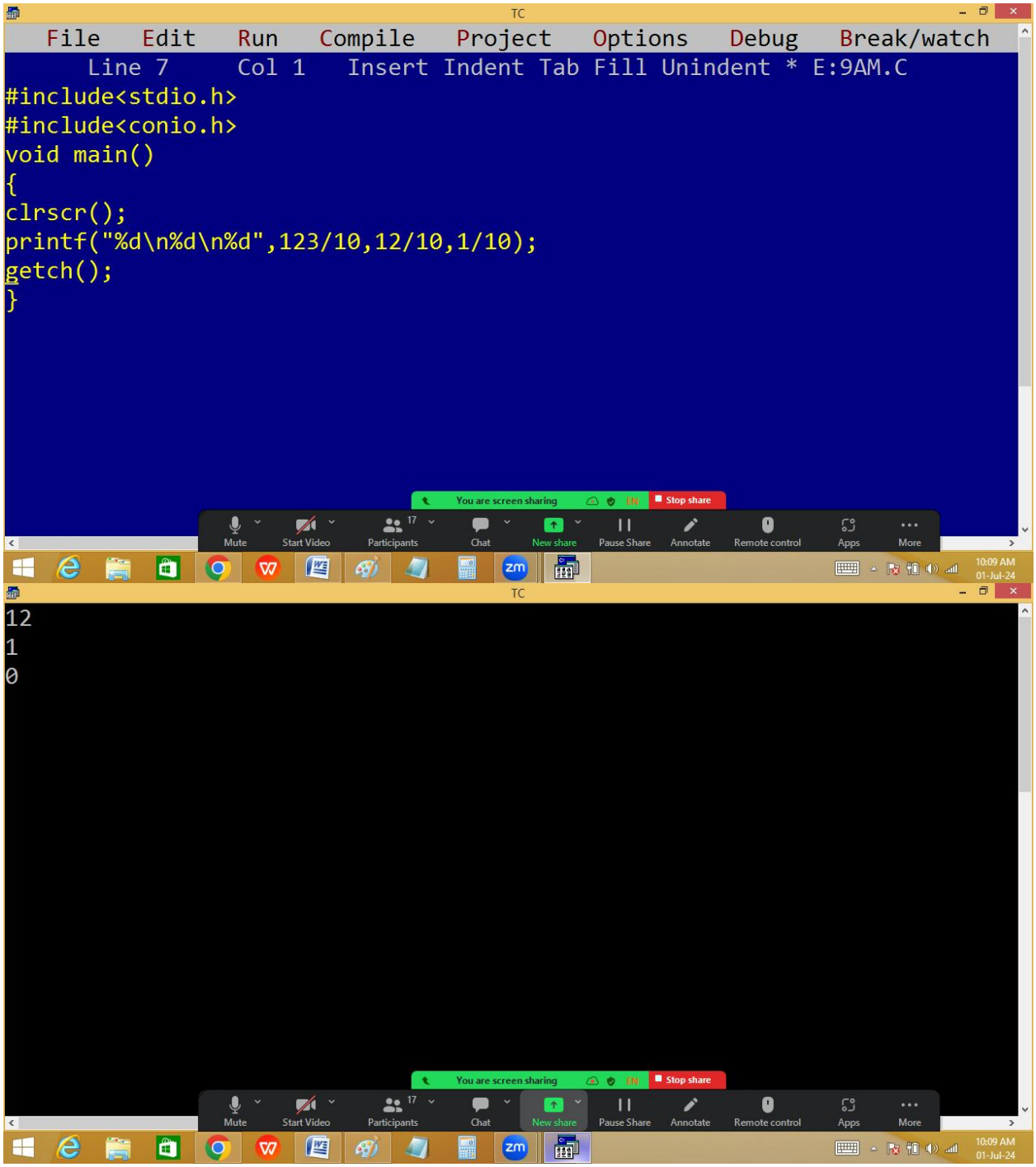


123/10=12

12/10=1

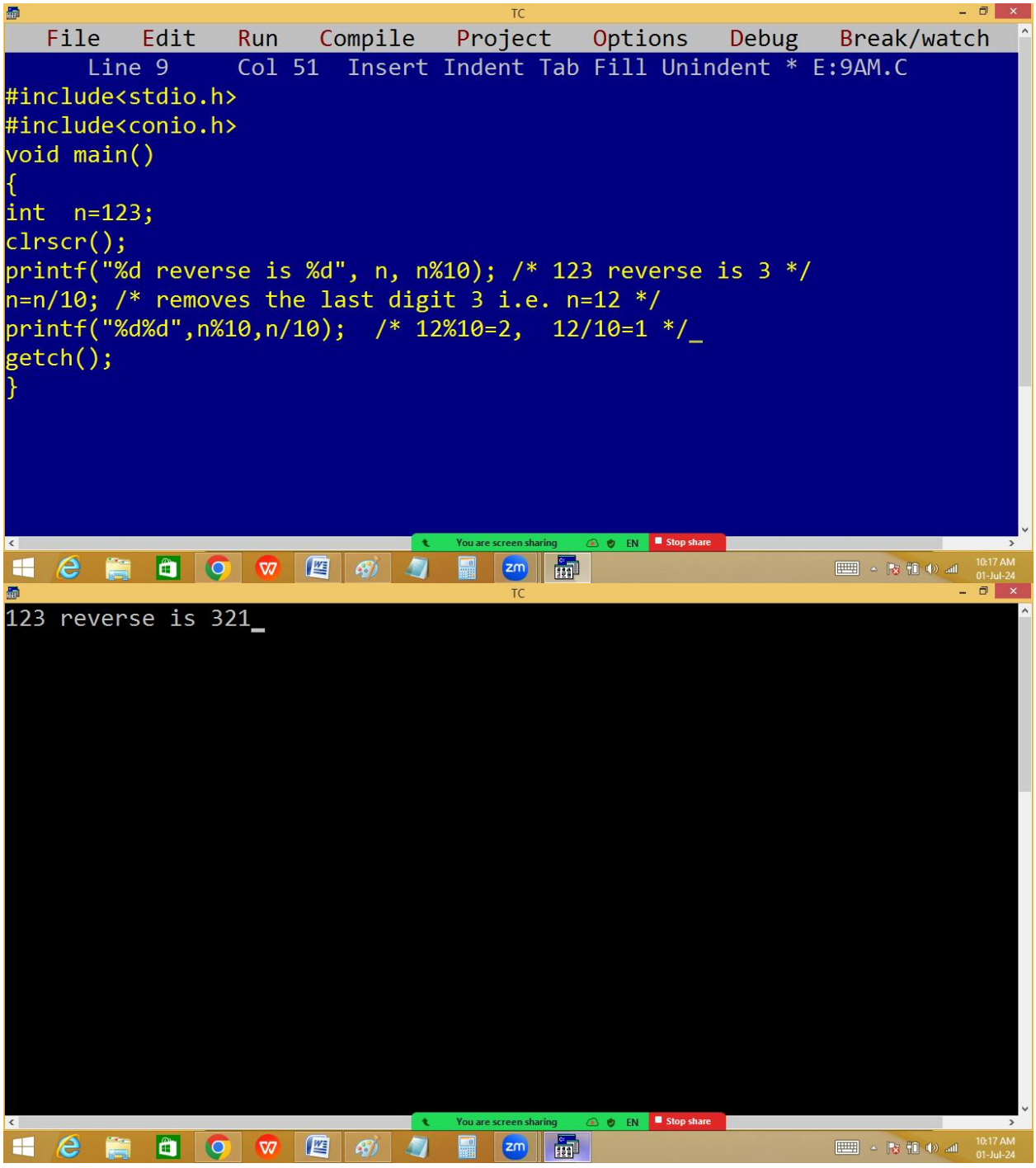
~~1~~/10=0

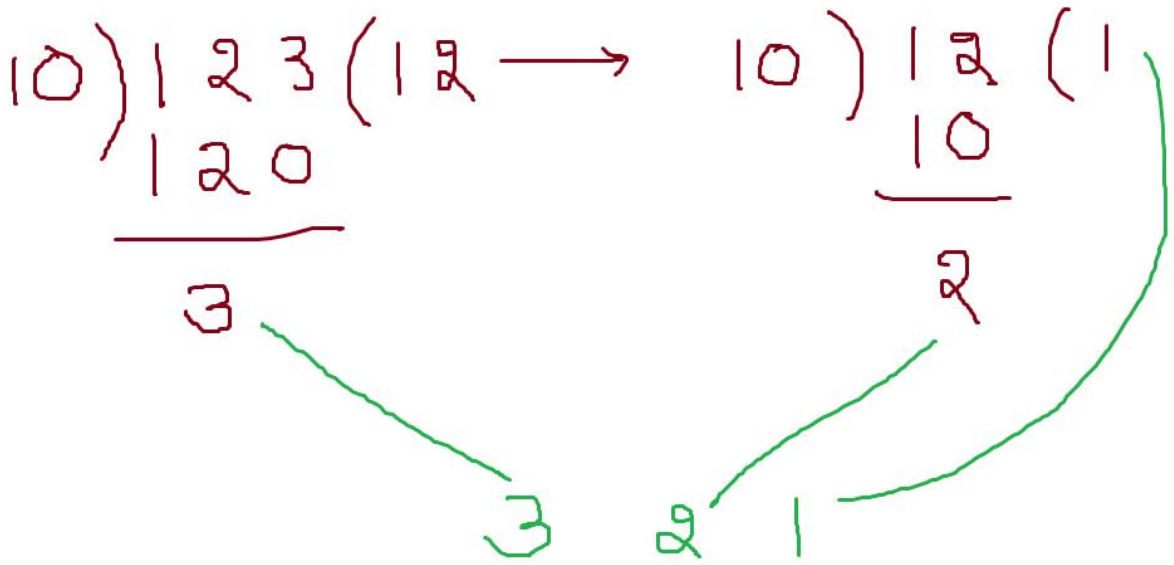
Note: Any no/10 removes the last digit.



Write a c program to print a 3 digit no in reverse order without using loop:

Eg: 123 reverse is 321


TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 51 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n=123;
clrscr();
printf("%d reverse is %d", n, n%10); /* 123 reverse is 3 */
n=n/10; /* removes the last digit 3 i.e. n=12 */
printf("%d%d",n%10,n/10); /* 12%10=2, 12/10=1 */
getch();
}
You are screen sharing
EN
Stop share
10:17 AM
01-Jul-24
TC
123 reverse is 321
You are screen sharing
EN
Stop share
10:17 AM
01-Jul-24

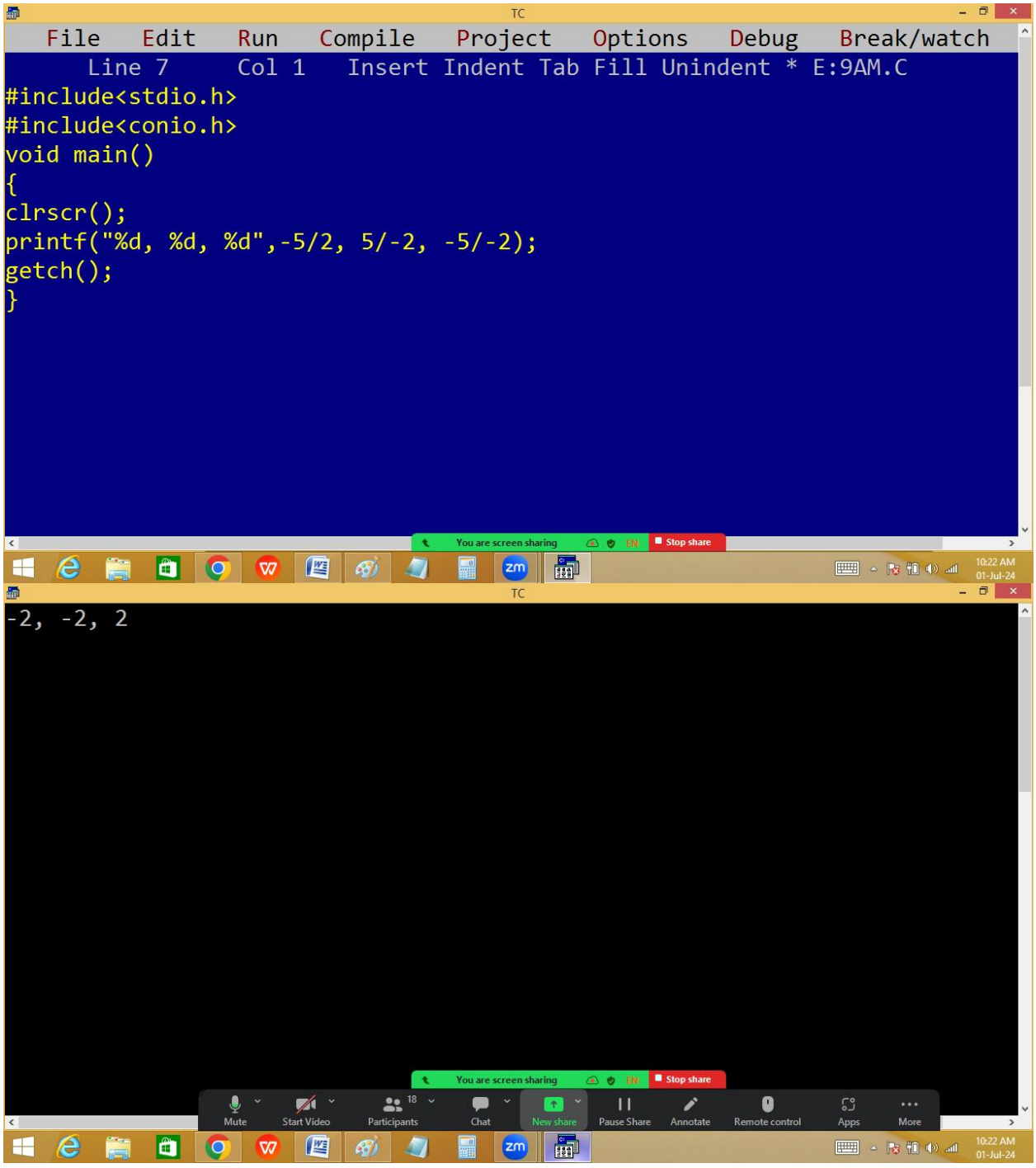


-5/2= -2

5/-2= -2

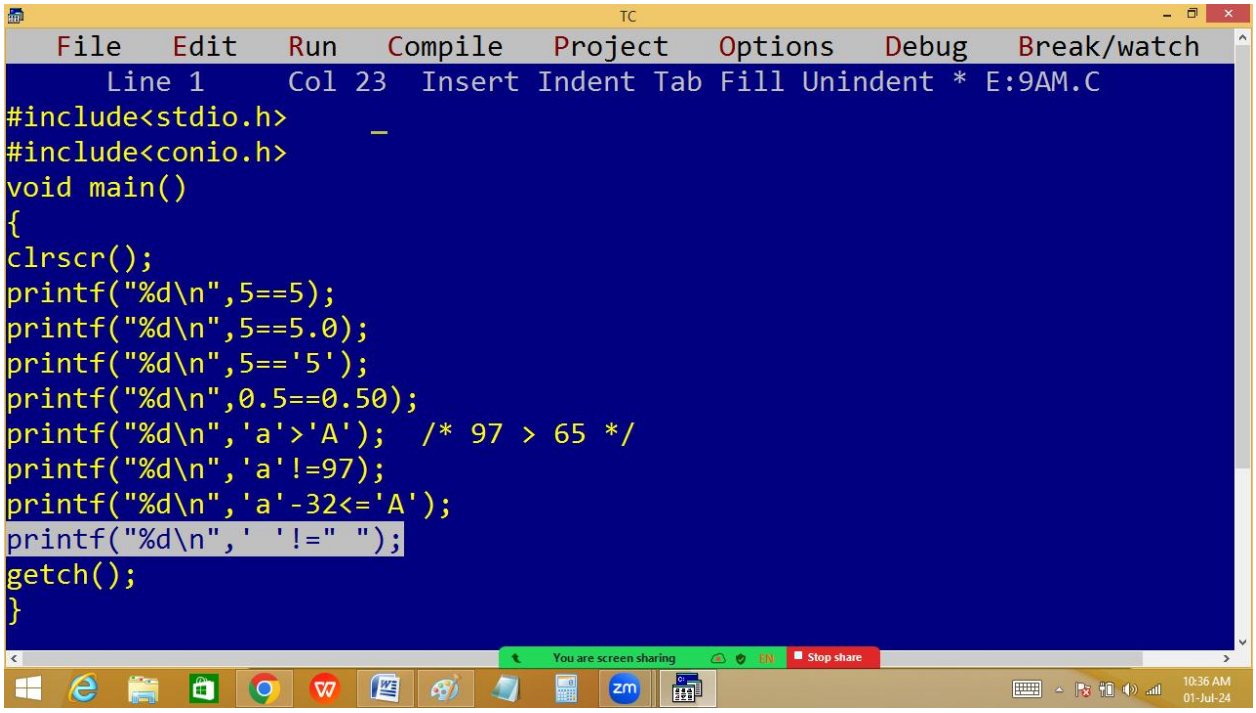
-5/-2= 2

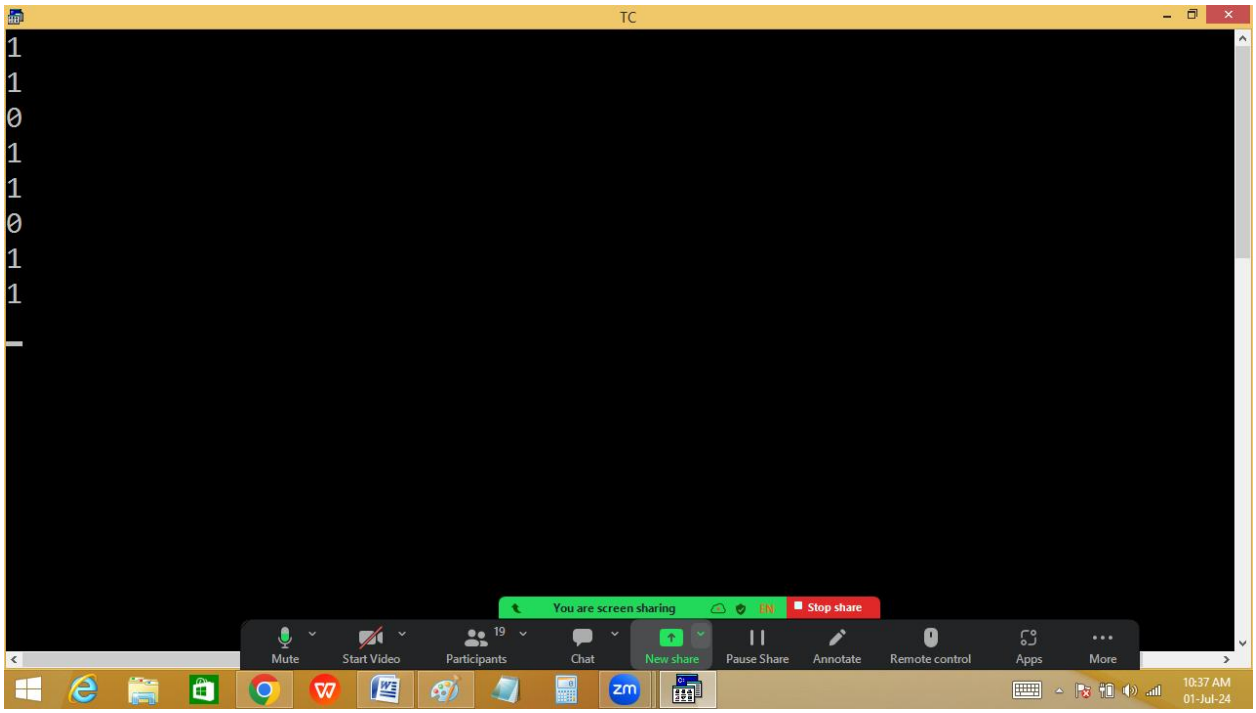
Note: In division any one operand is negative then result also negative. If both are negative then result is positive.

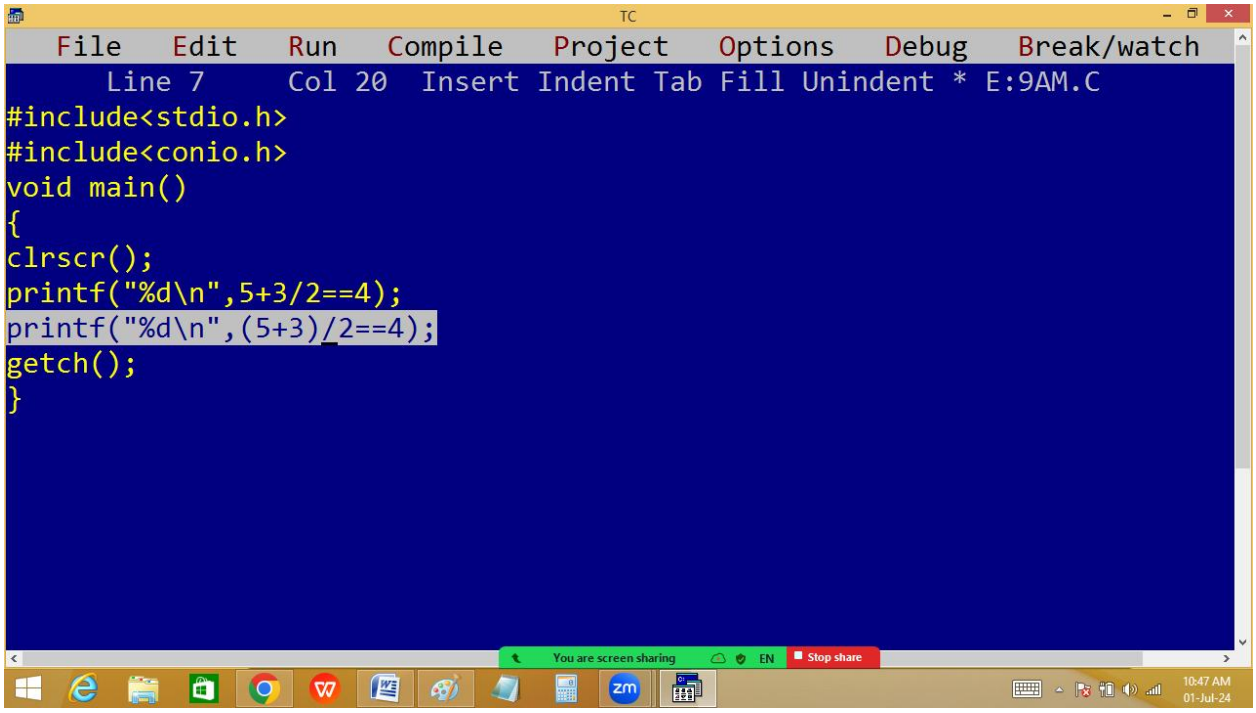

TC
File Edit Run Compile Project Options Debug Break/watch
Line 7 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d, %d, %d",-5/2, 5/-2, -5/-2);
getch();
}
You are screen sharing
Stop share
TC
-2, -2, 2
You are screen sharing
Stop share
Mute
Start Video
Participants
Chat
New share
Pause Share
Annotate
Remote control
Apps
More

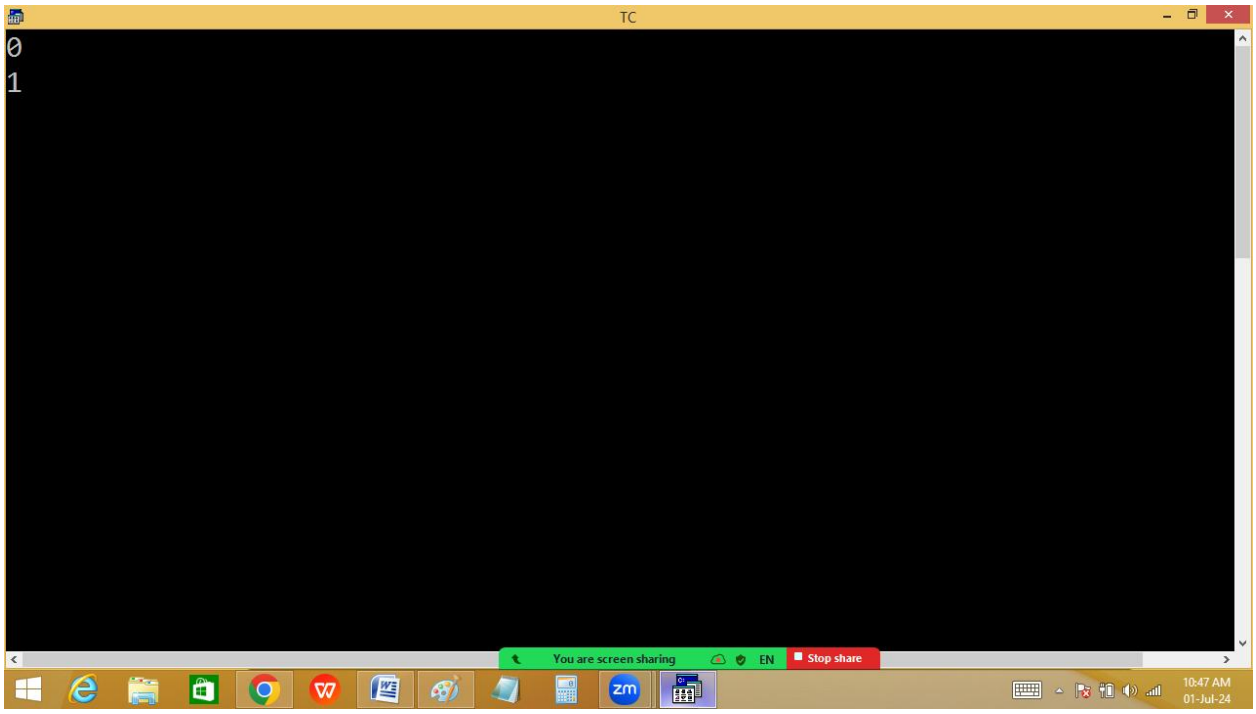
Relational operators [== (comparison), <, >, <=, >=, != (not equal)]:

They are used to check the given condition or expression is true or false. If condition true always it return 1 and false it return 0.


TC
File Edit Run Compile Project Options Debug Break/watch
Line 1 Col 23 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n",5==5);
printf("%d\n",5==5.0);
printf("%d\n",5=='5');
printf("%d\n",0.5==0.50);
printf("%d\n",'a'>'A'); /* 97 > 65 */
printf("%d\n",'a'!=97);
printf("%d\n",'a'-32<='A');
printf("%d\n",' '!=" ");
getch();
}
You are screen sharing
Stop share
10:36 AM
01-Jul-24


TC
1
1
0
1
1
0
1
1
You are screen sharing
Stop share
Mute
Start Video
Participants
19
Chat
New share
Pause Share
Annotate
Remote control
Apps
More
10:37 AM
01-Jul-24





Operator precedence / Operator priority

(ASSOCIATION OF OPERATORS)

1. ()
2. +, - , ! (sign operators, unary operators)
3. ++, -- (pre increment & decrement)
4. *, / , %
5. +, - (Binary)
6. = =, !=
7. &&
8. ||
9. **? :** (ternary operator)
10. =
11. ++, -- (Post increment & decrement)
12. , (comma)