OPERATORS

Operator is a special symbol used to do a particular task [work]. C comes with 44 operators and 14 separators.

Operator works on operands. Based on **no of operands participating in operation, the operators divided into 3 types**.

1. **Unary operator**: Require one operand.
   Eg: +a, -a, a++, a--, ++a, --a, sizeof(a), ~a, !a,…
2. **Binary operators**: Require two operands.
   Eg: a+b, a>b, a!=b, a==b, a<<b,….

3. **Ternary / Conditional operator[?:]**
   Require three operands.
   Eg: a>b ?"a big":"b big";

Based on operation, the operators divided into several types.

1. **Assignment operator [=]**: It copies the value on its right side into the variable on its left side. In assignment operation, **the left side operand should be a variable** i.e. constant or expressions not allowed on left side.
   Eg:
   a=10;
   b=1.2;
   c='X';

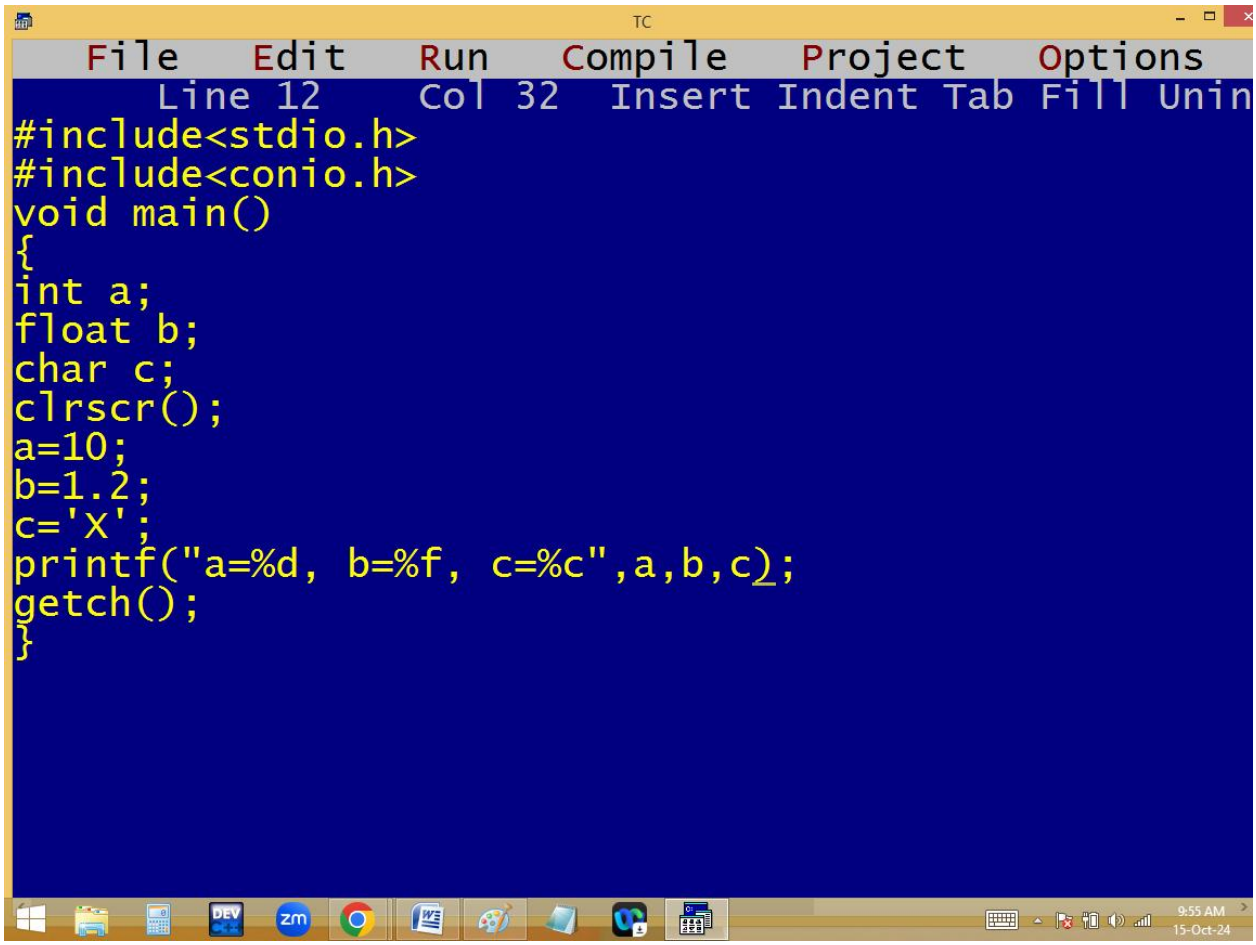
d=”abc”; ➔ Lvalue error

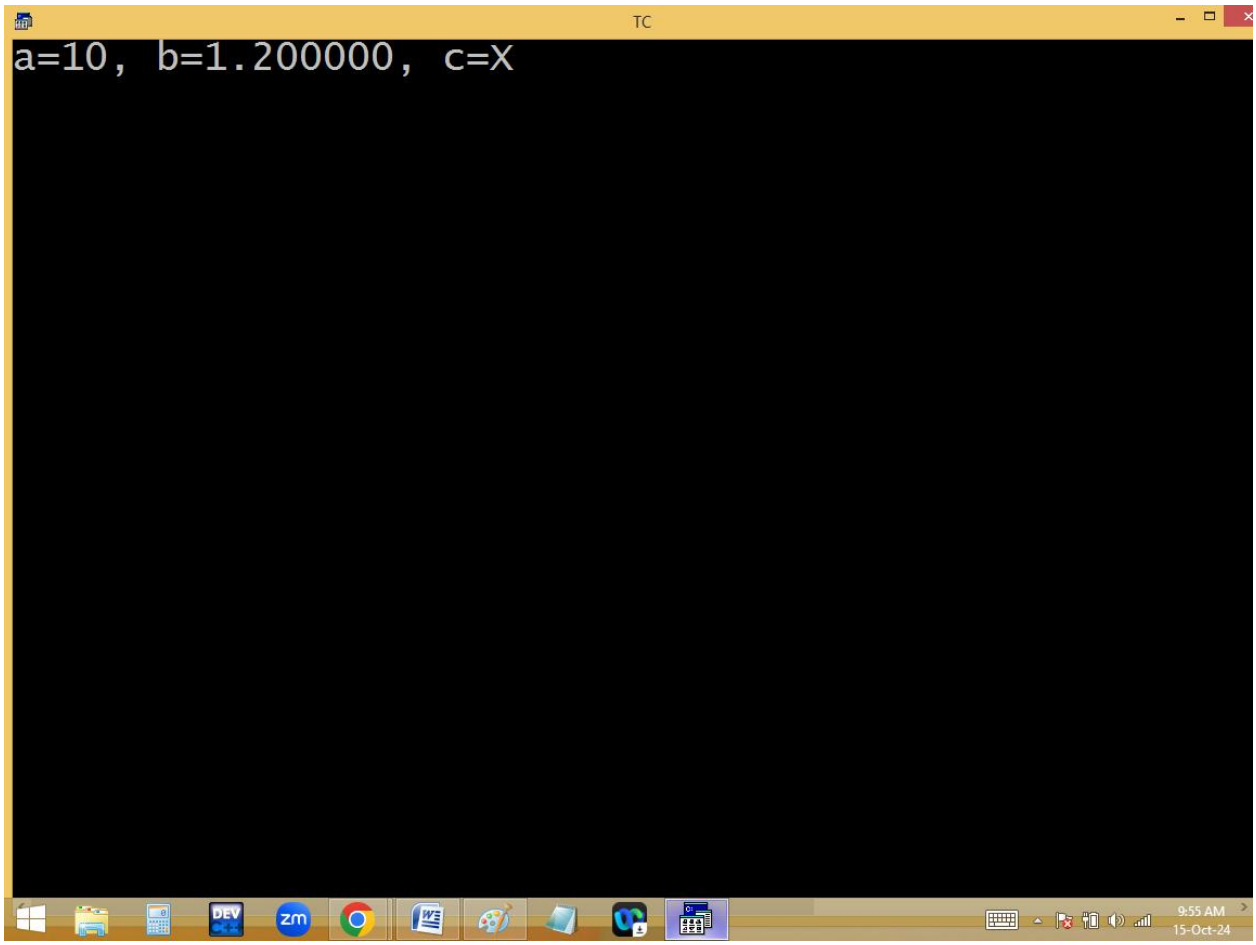
10=20; ➔ Lvalue error

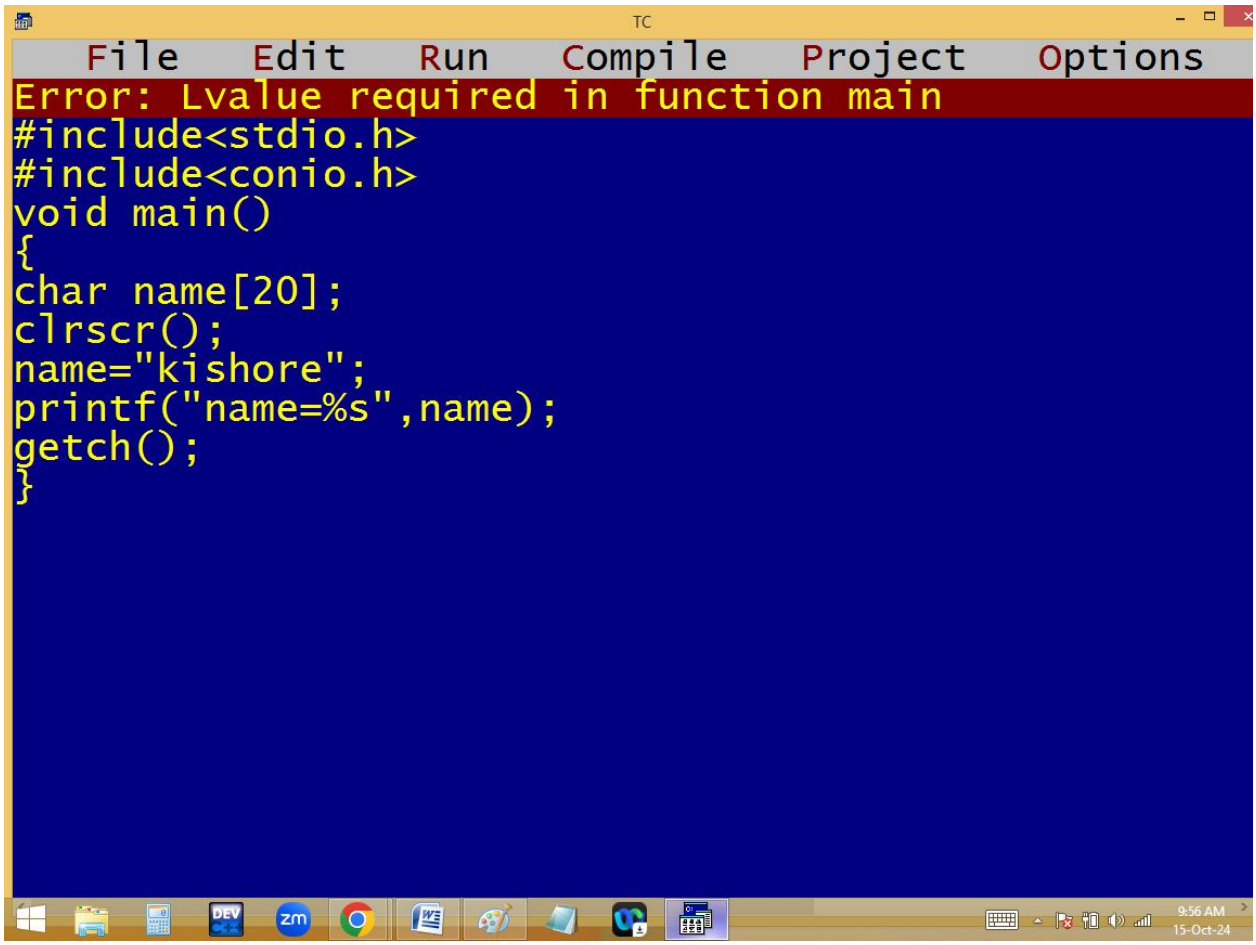
c=a+b;

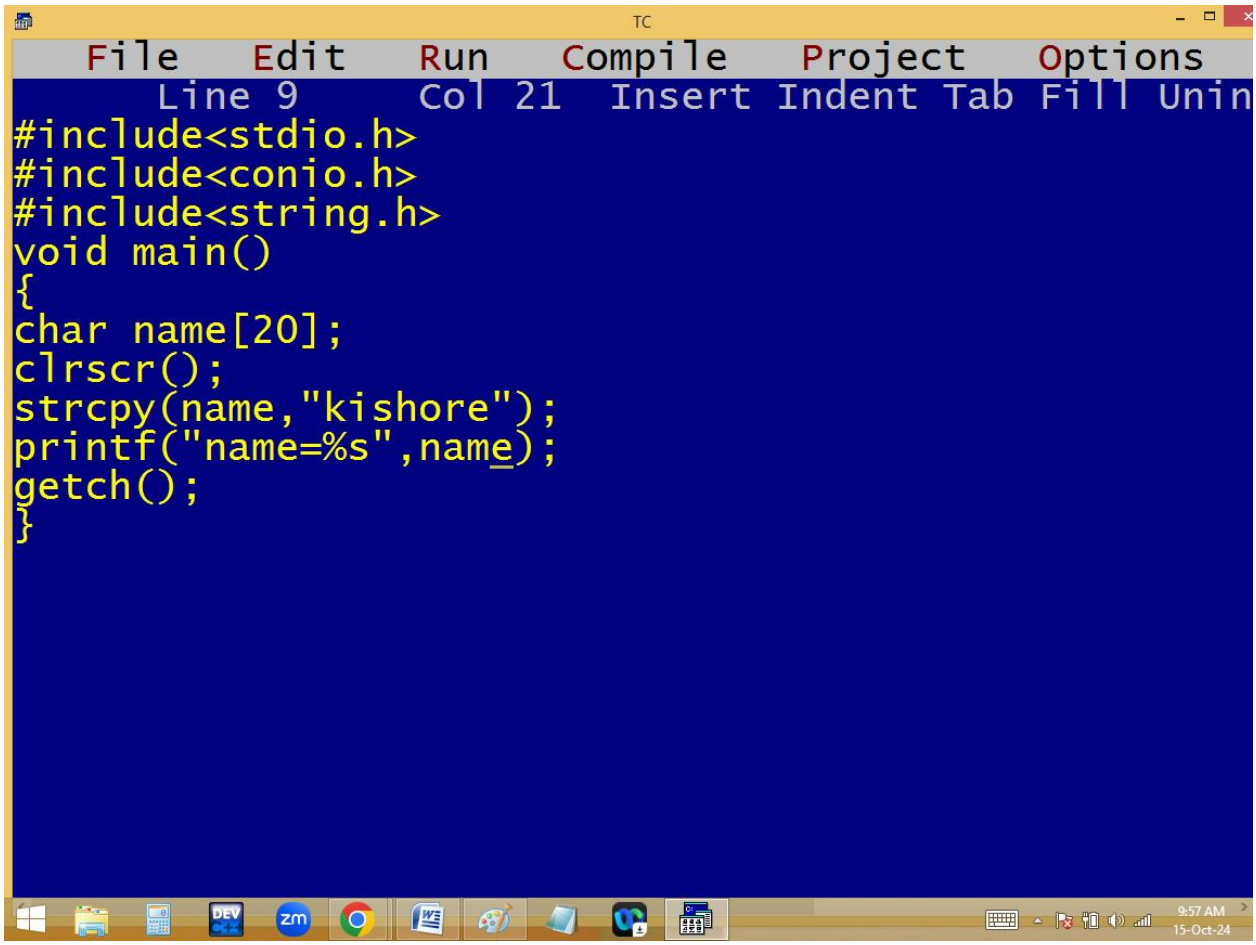
a+b=c; ➔ Lvalue error

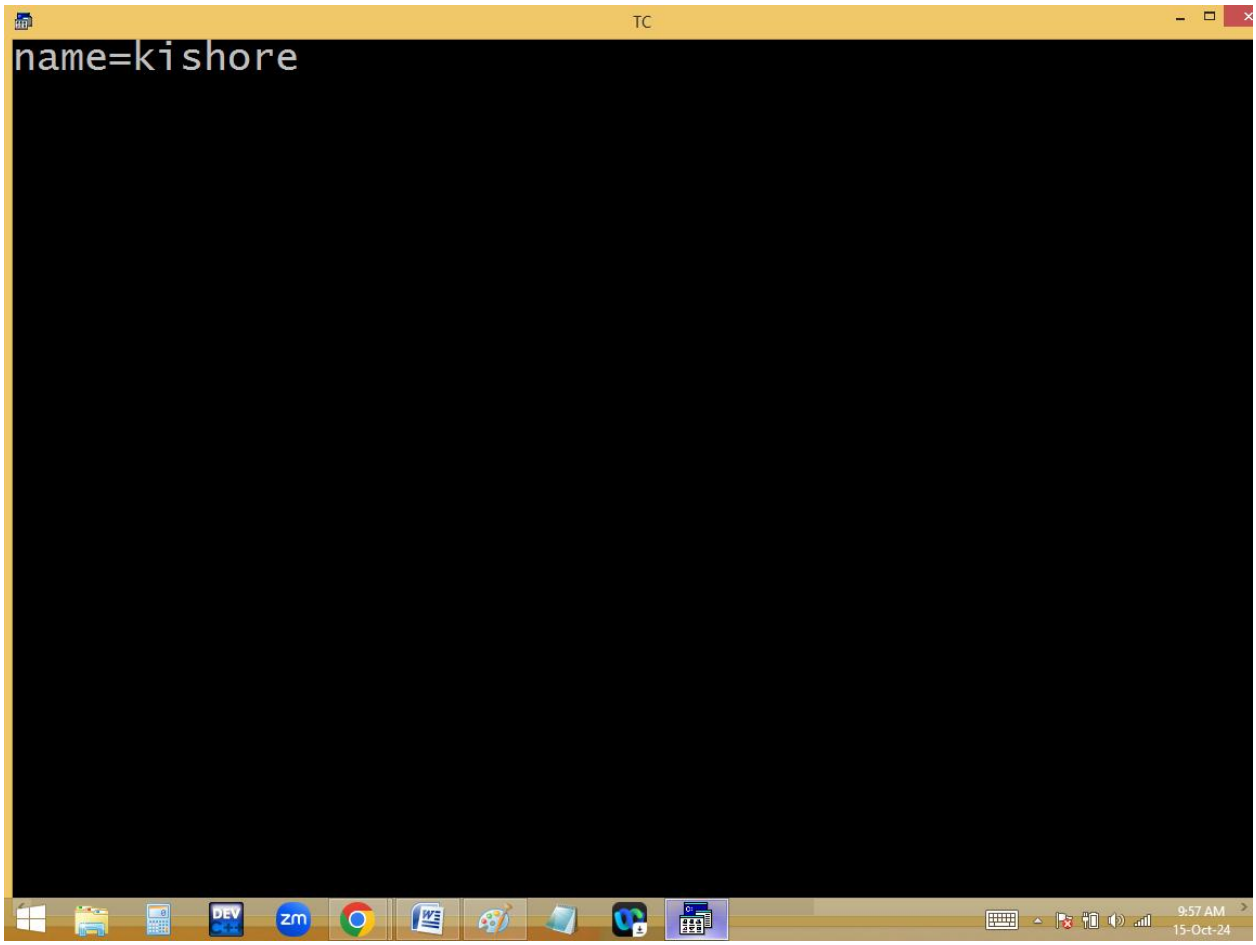
a=b=c=100;

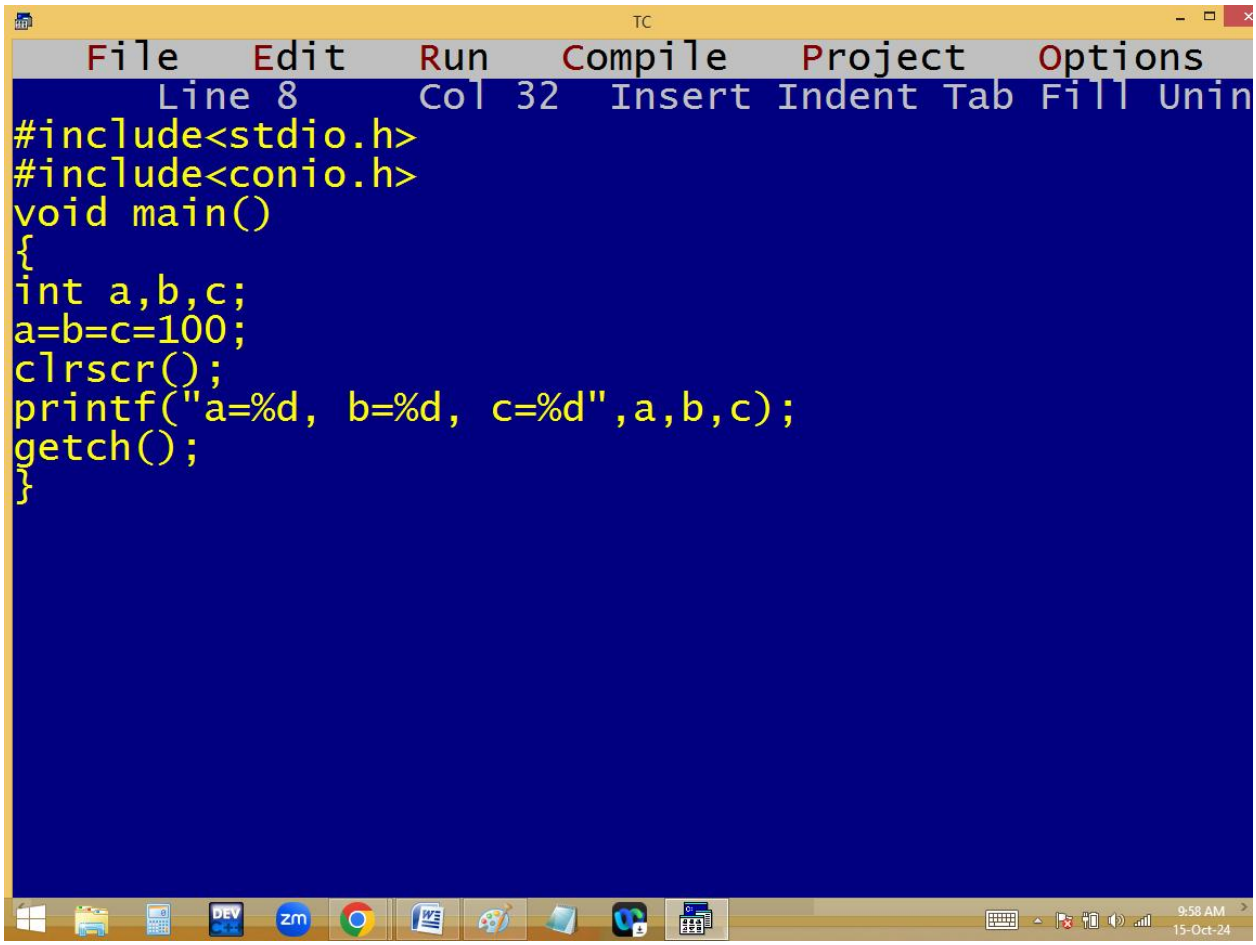

TC
File Edit Run Compile Project Options
Line 12 Col 32 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
int a;
float b;
char c;
clrscr();
a=10;
b=1.2;
c='X';
printf("a=%d, b=%f, c=%c",a,b,c);
getch();
}
9:55 AM
15-Oct-24

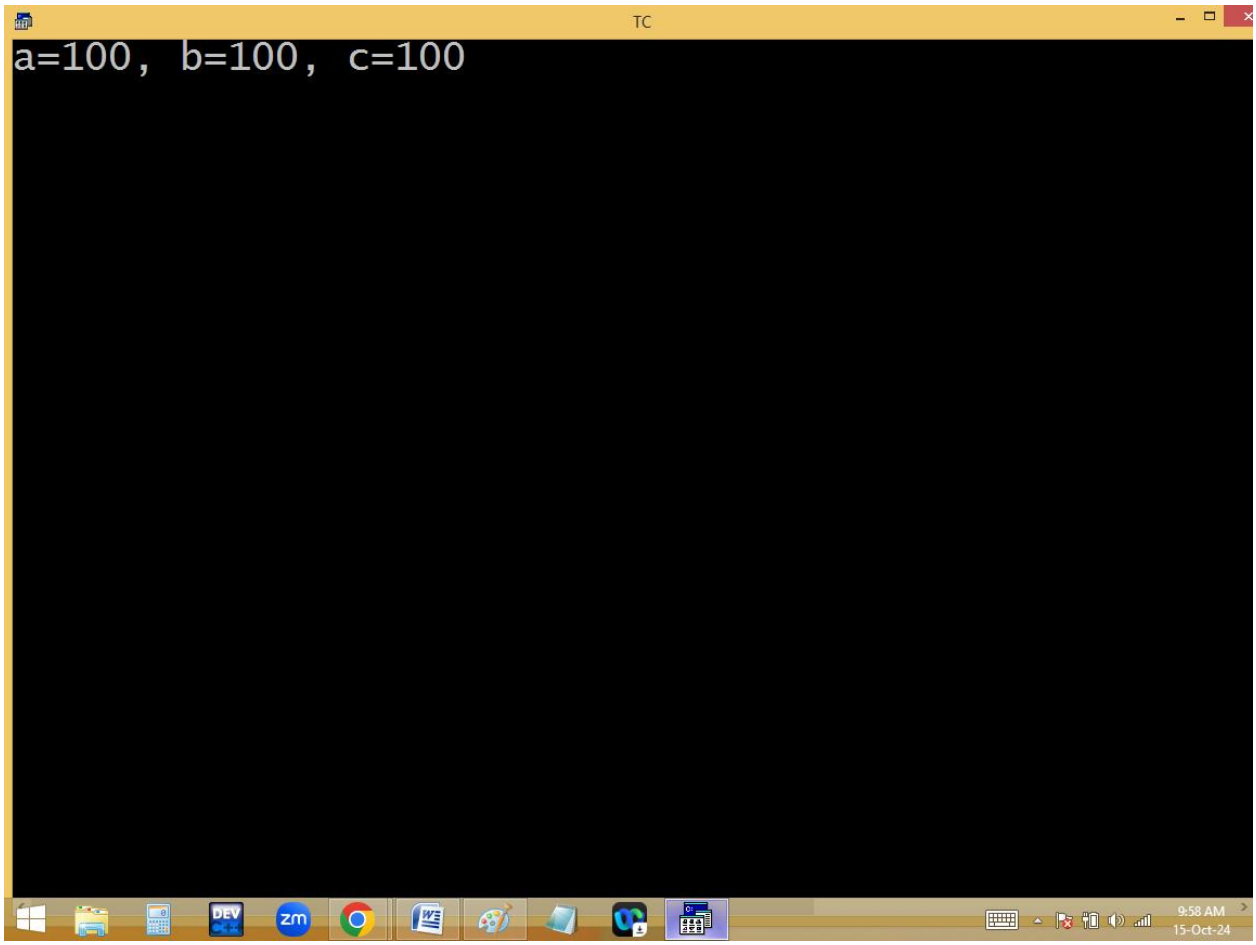

TC
a=10, b=1.200000, c=X
9:55 AM
15-Oct-24

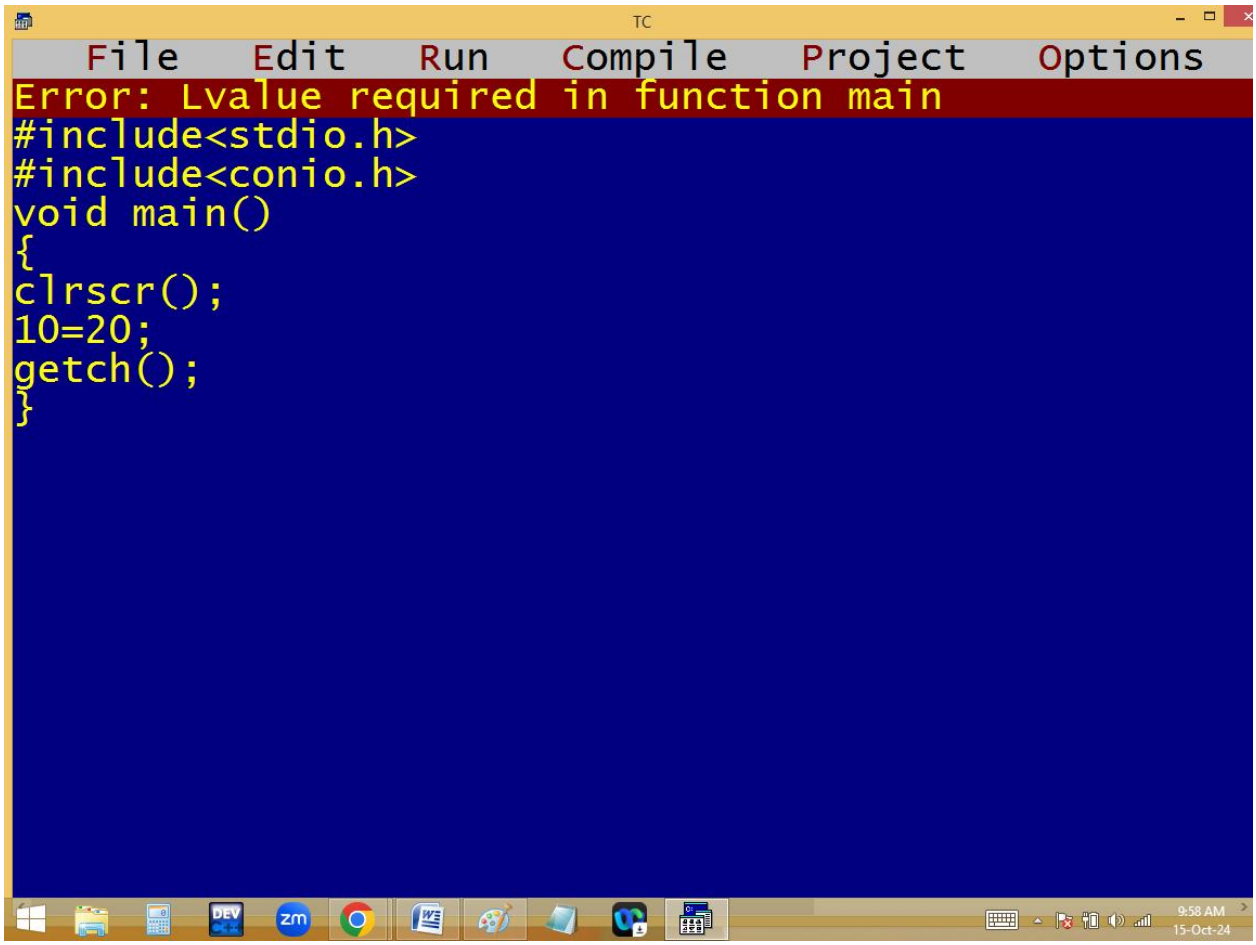

TC
File Edit Run Compile Project Options
Error: Lvalue required in function main
#include<stdio.h>
#include<conio.h>
void main()
{
char name[20];
clrscr();
name="kishore";
printf("name=%s",name);
getch();
}
9:56 AM
15-Oct-24

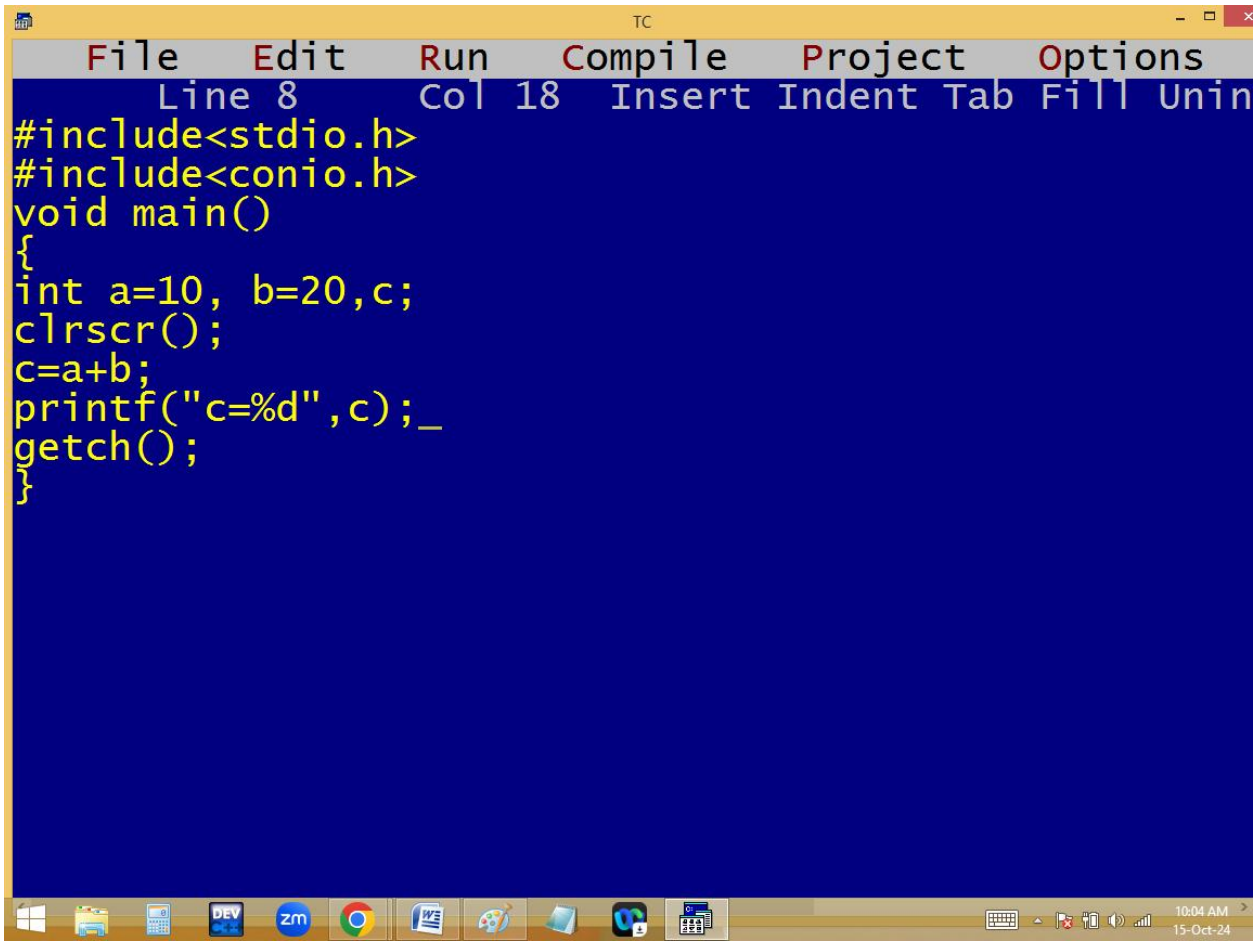

TC
File Edit Run Compile Project Options
Line 9 Col 21 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char name[20];
clrscr();
strcpy(name,"kishore");
printf("name=%s",name);
getch();
}
9:57 AM
15-Oct-24

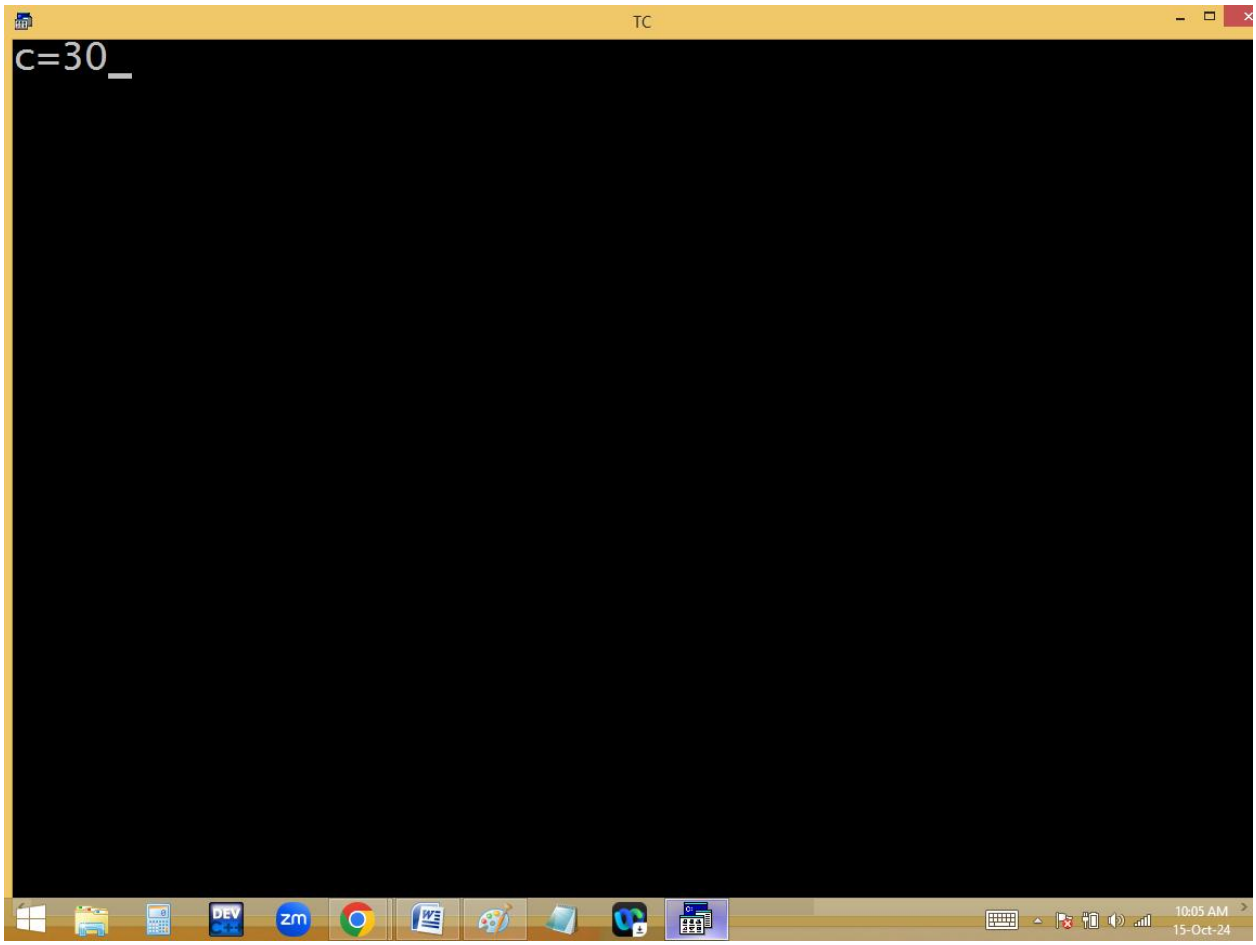

TC
name=kishore
9:57 AM
15-Oct-24

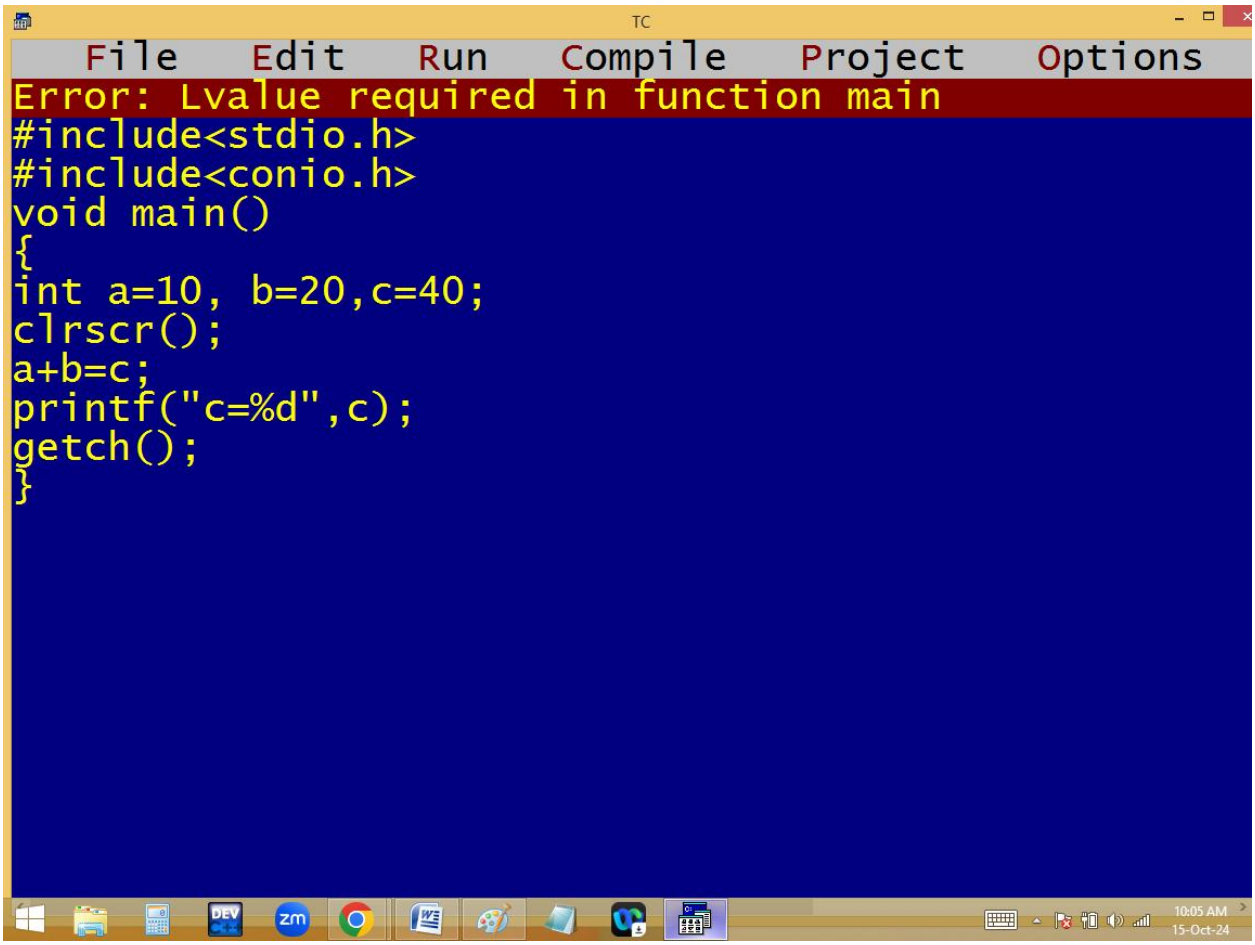

TC
File Edit Run Compile Project Options
Line 8 Col 32 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
a=b=c=100;
clrscr();
printf("a=%d, b=%d, c=%d",a,b,c);
getch();
}
9:58 AM
15-Oct-24


TC
a=100, b=100, c=100
9:58 AM
15-Oct-24


TC
File Edit Run Compile Project Options
Error: Lvalue required in function main
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
10=20;
getch();
}
9:58 AM
15-Oct-24


TC
File Edit Run Compile Project Options
Line 8 Col 18 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, b=20,c;
clrscr();
c=a+b;
printf("c=%d",c);_
getch();
}
10:04 AM
15-Oct-24


TC
c=30_
10:05 AM
15-Oct-24



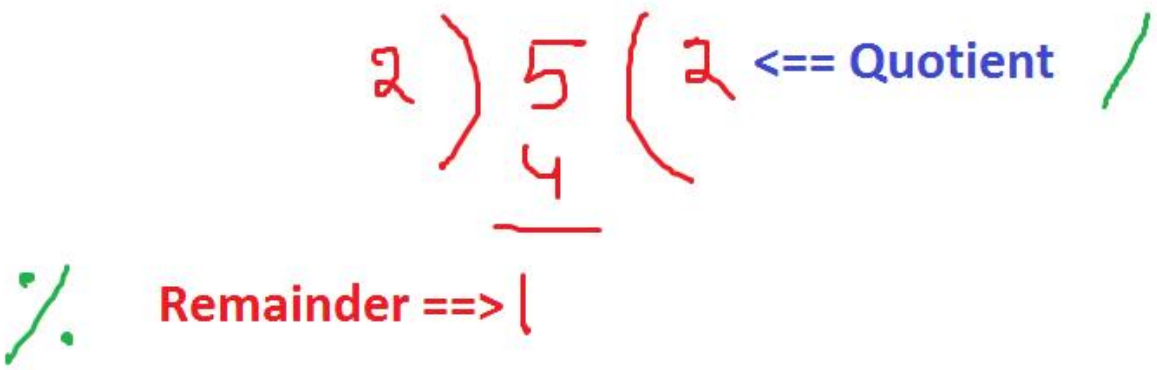
2. **Arithmetic operators [+, -, *, %, /]:**

   They are used to perform mathematical calculations.

   Eg: a+b, a-b, a*b, a%b, a/b,.....

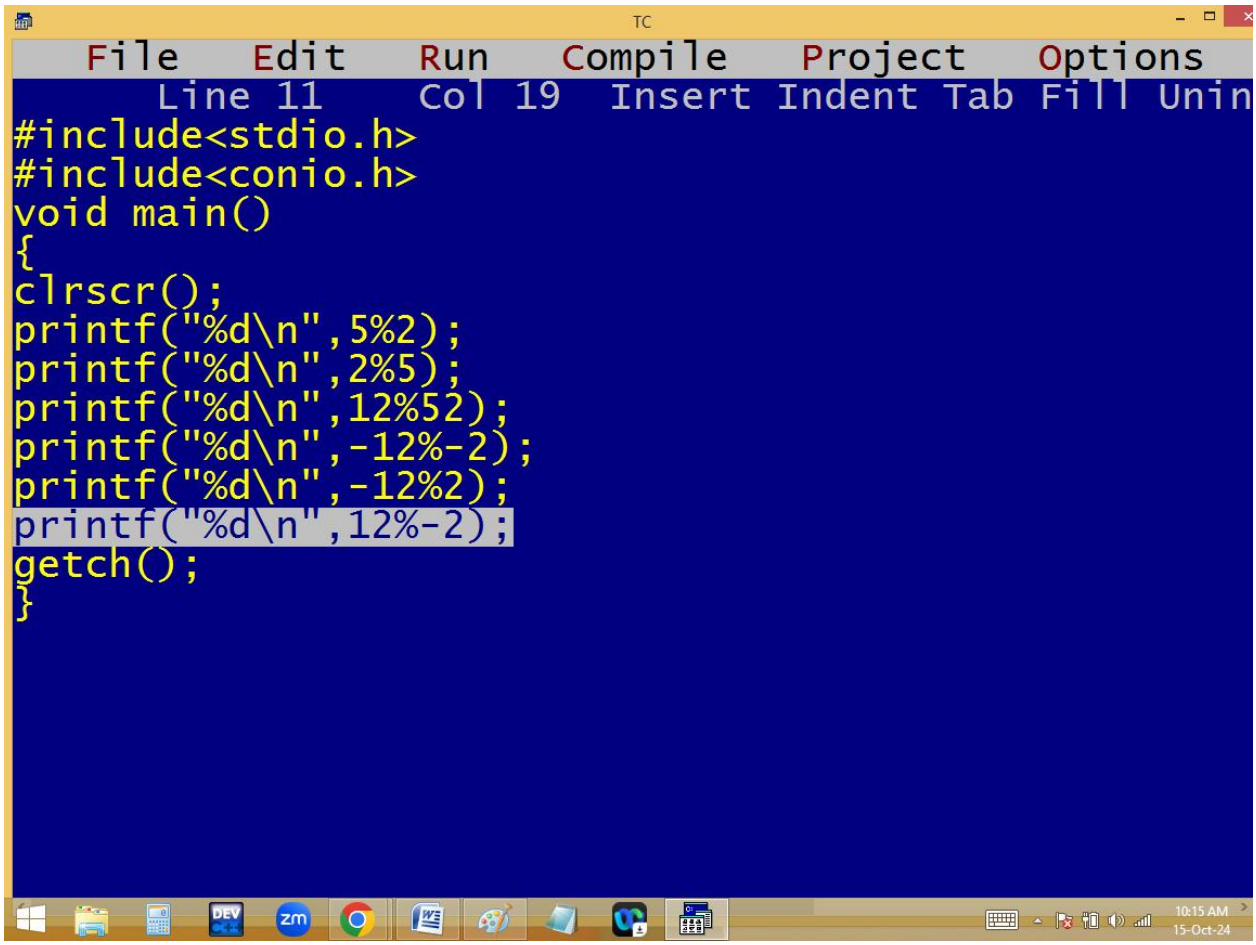
% - modules [Remainder]:

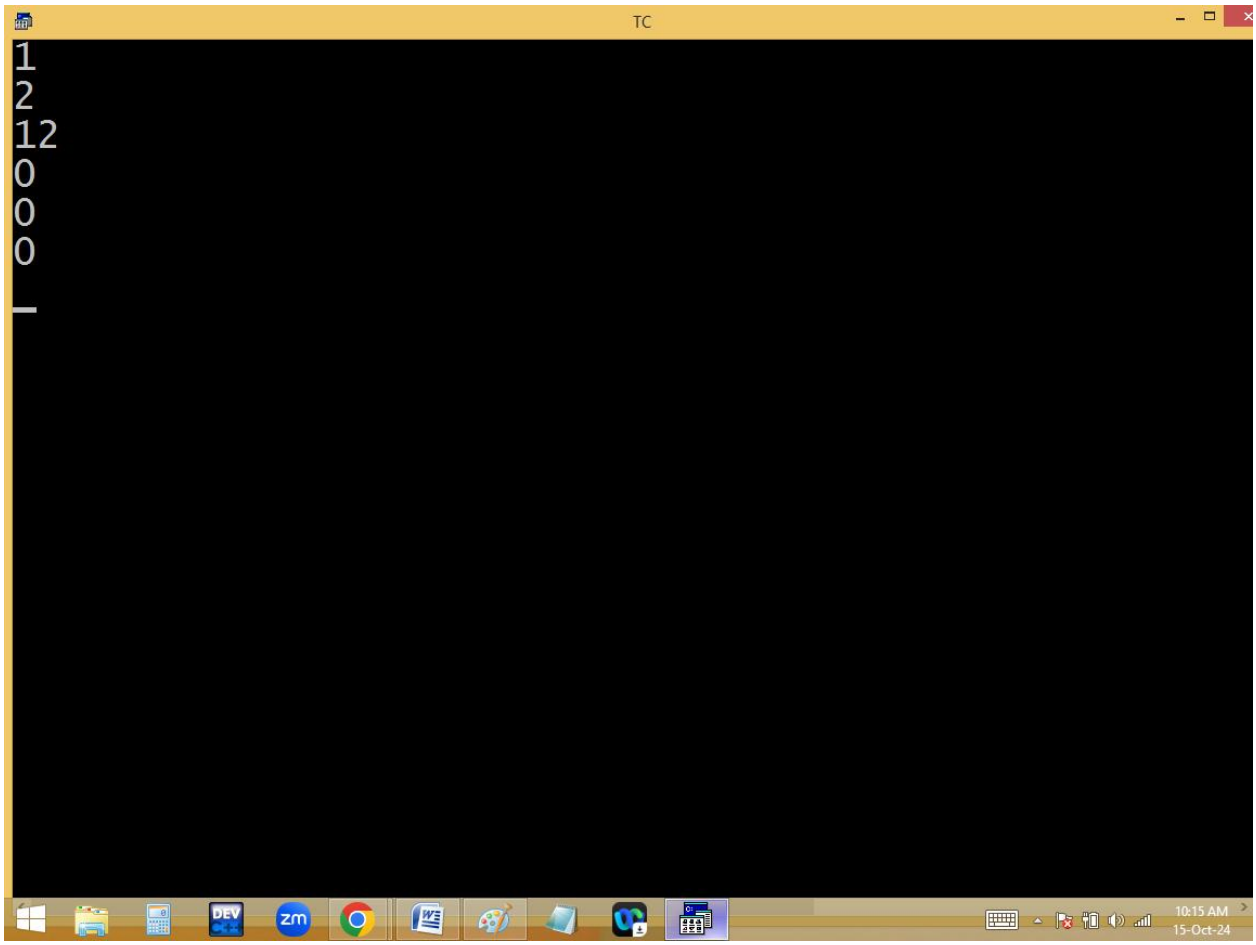
5%2=1



2%5=2

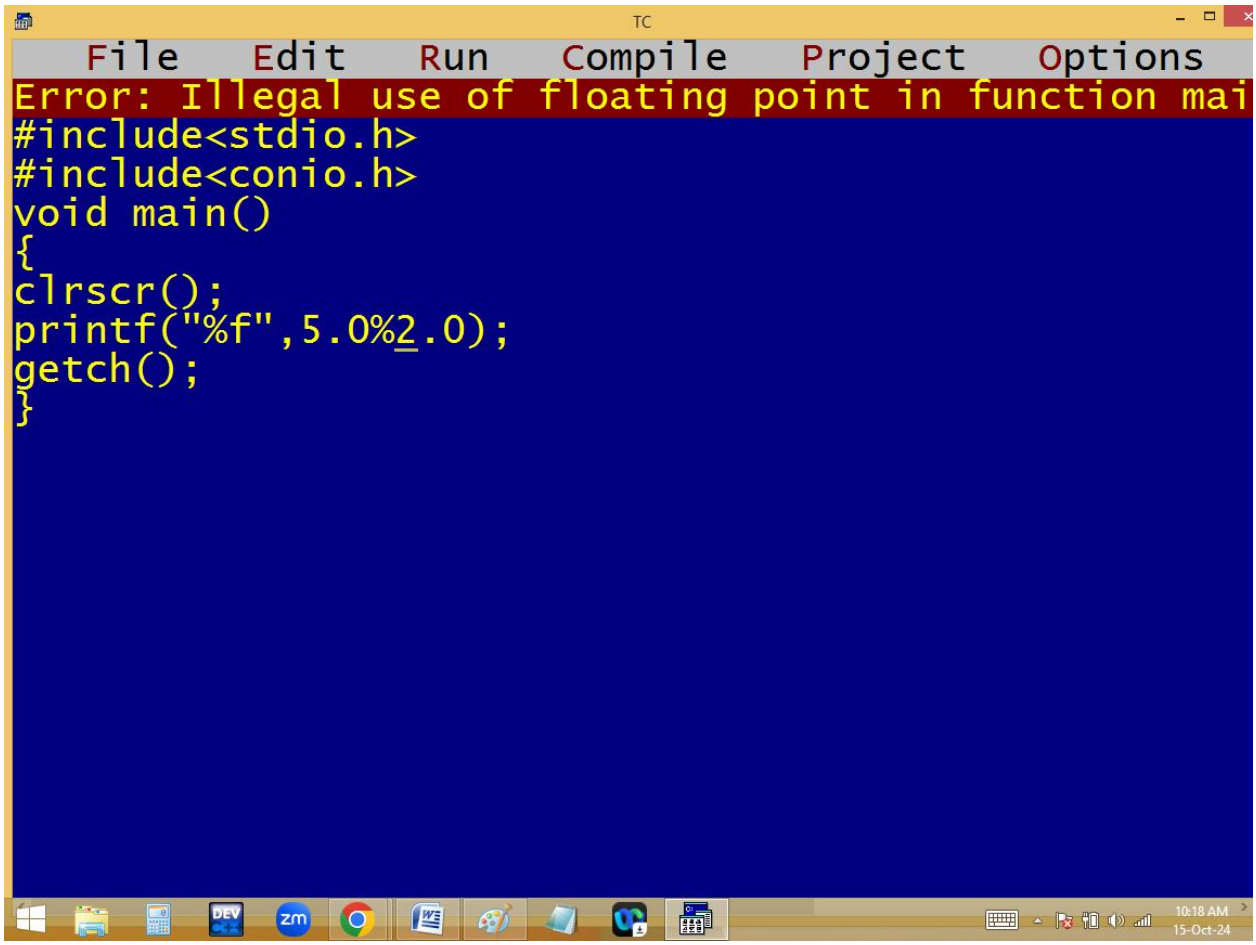
Note: If the divisor bigger than dividend then dividend is the answer.

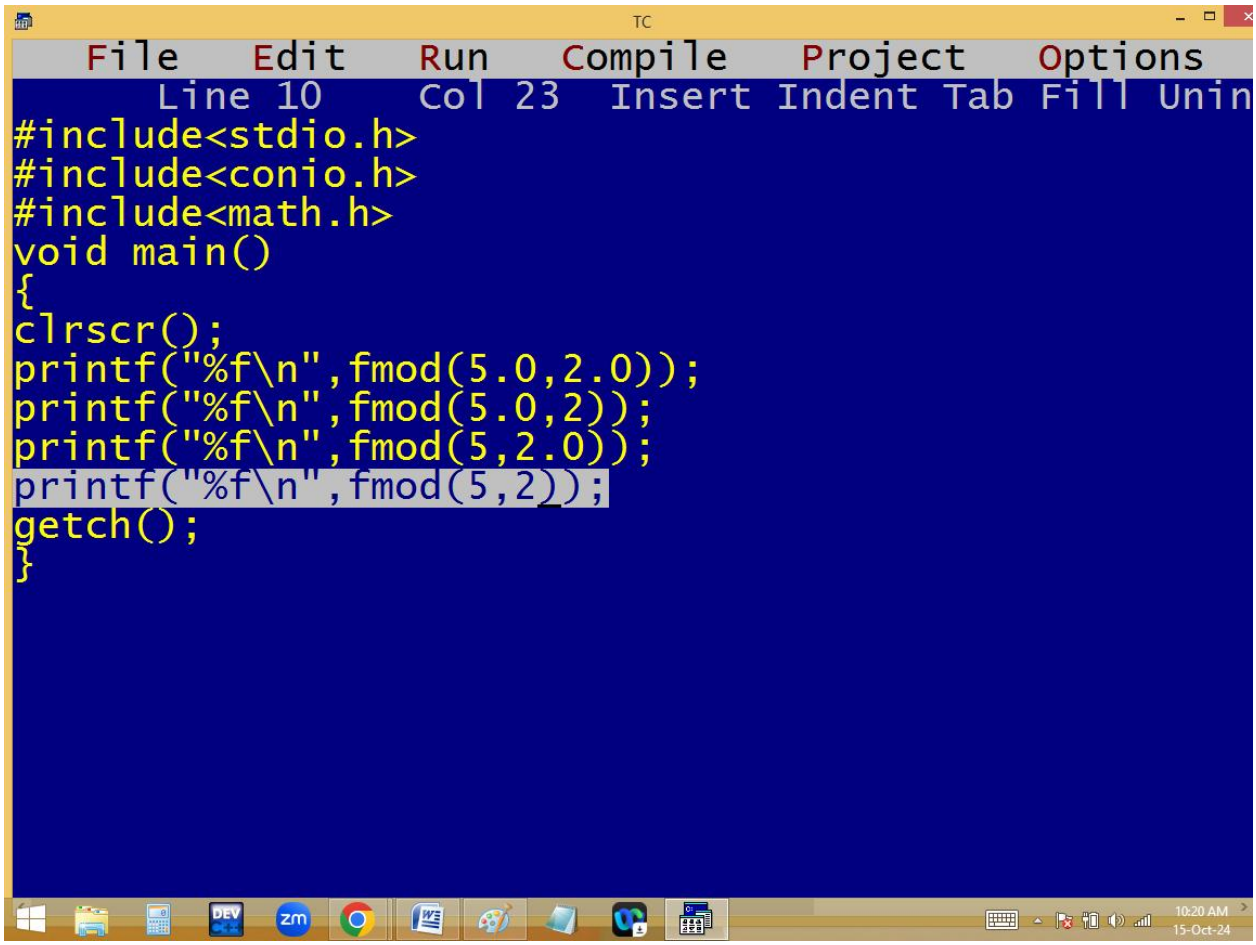

TC
File Edit Run Compile Project Options
Line 11 Col 19 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n",5%2);
printf("%d\n",2%5);
printf("%d\n",12%52);
printf("%d\n",-12%-2);
printf("%d\n",-12%2);
printf("%d\n",12%-2);
getch();
}
10:15 AM
15-Oct-24

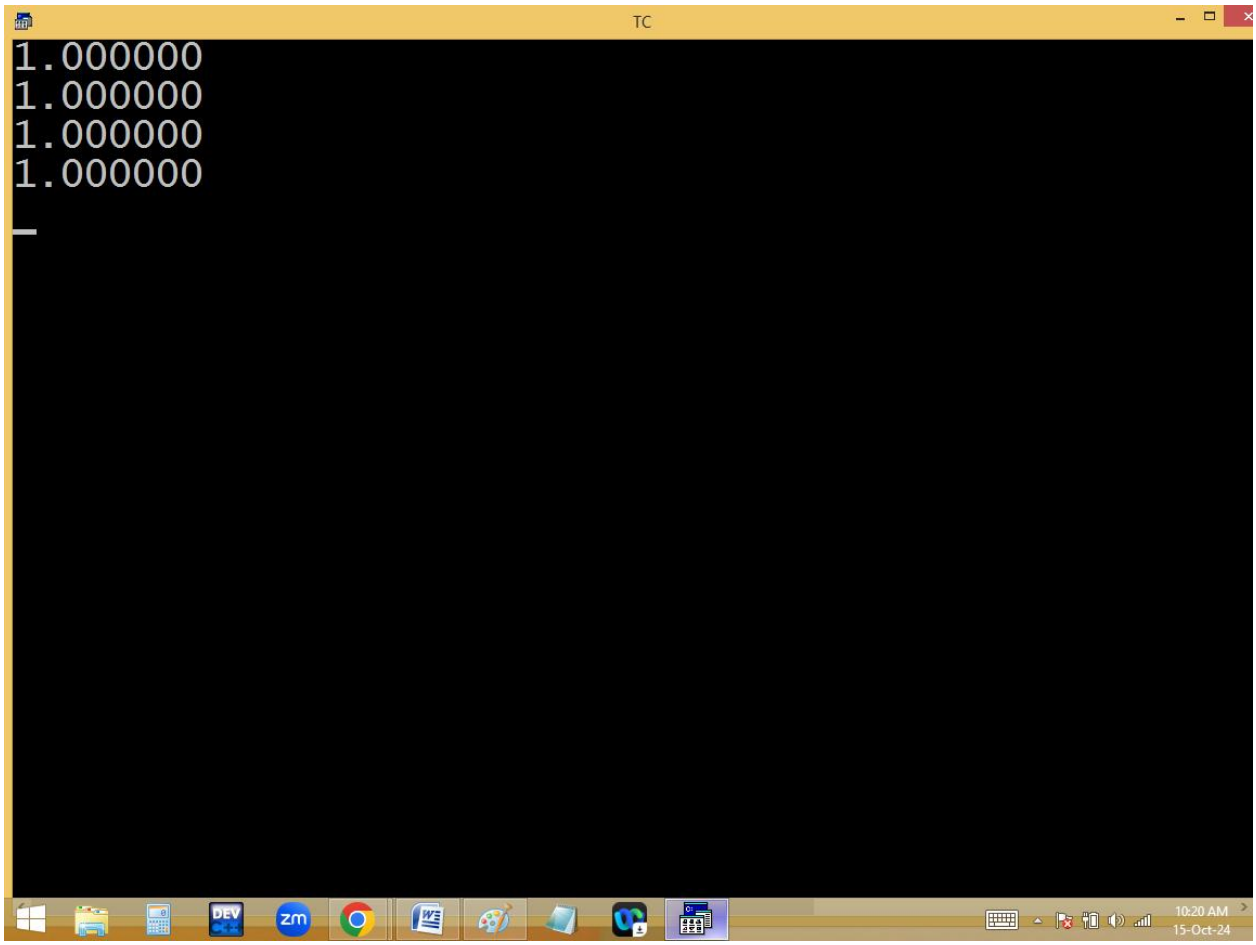


5.0 % 2.0 = Error

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


TC
File Edit Run Compile Project Options
Error: Illegal use of floating point in function mai
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%f",5.0%2.0);
getch();
}
10:18 AM
15-Oct-24


TC
File Edit Run Compile Project Options
Line 10 Col 23 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
#include<math.h>
void main()
{
clrscr();
printf("%f\n",fmod(5.0,2.0));
printf("%f\n",fmod(5.0,2));
printf("%f\n",fmod(5,2.0));
printf("%f\n",fmod(5,2));
getch();
}
10:20 AM
15-Oct-24

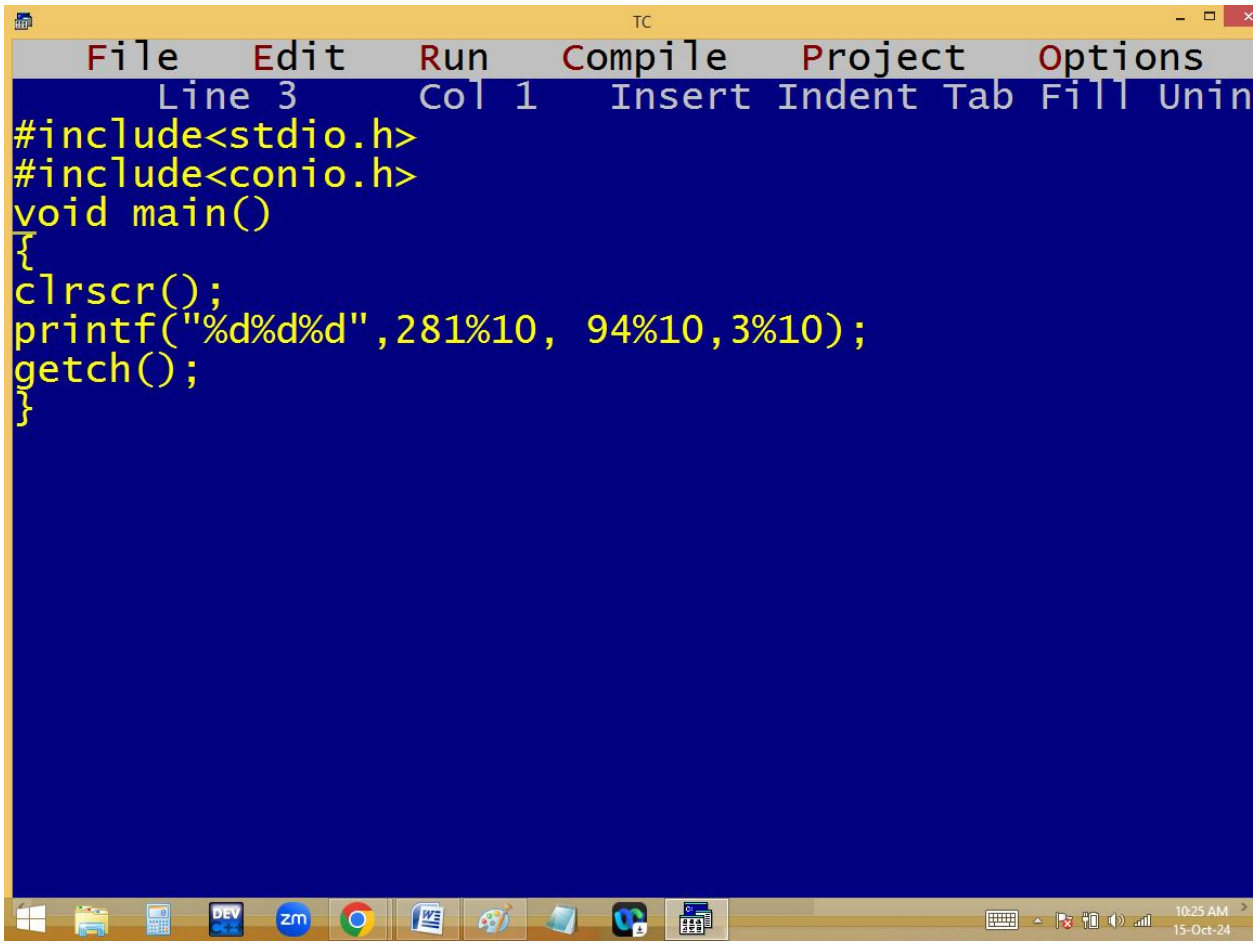


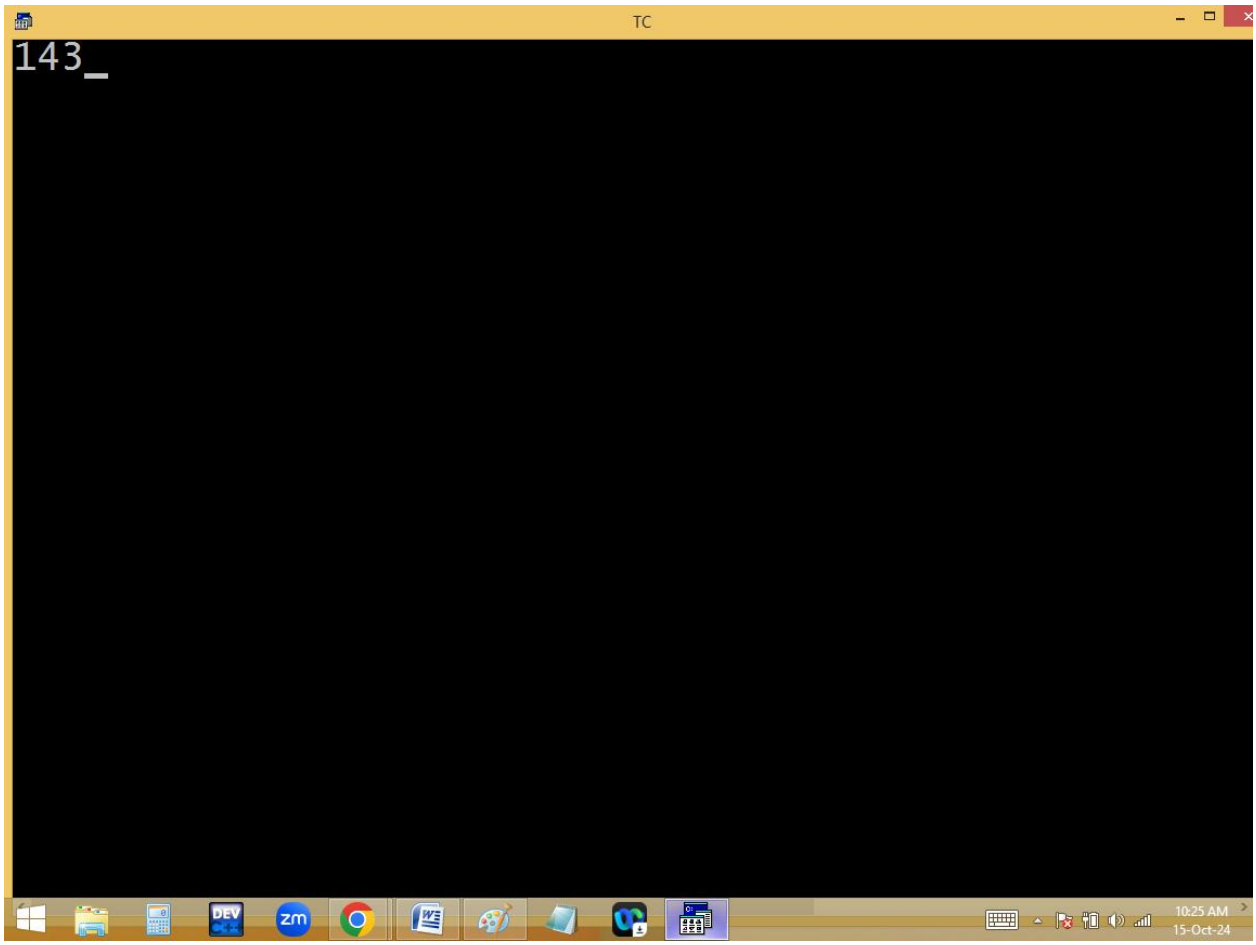
281 % 10 = 1

94 % 10 = 4

3 % 10 = 3

Note: Any no%10 gives last digit.


TC
File Edit Run Compile Project Options
Line 3 Col 1 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d%d%d",281%10, 94%10,3%10);
getch();
}
10:25 AM
15-Oct-24

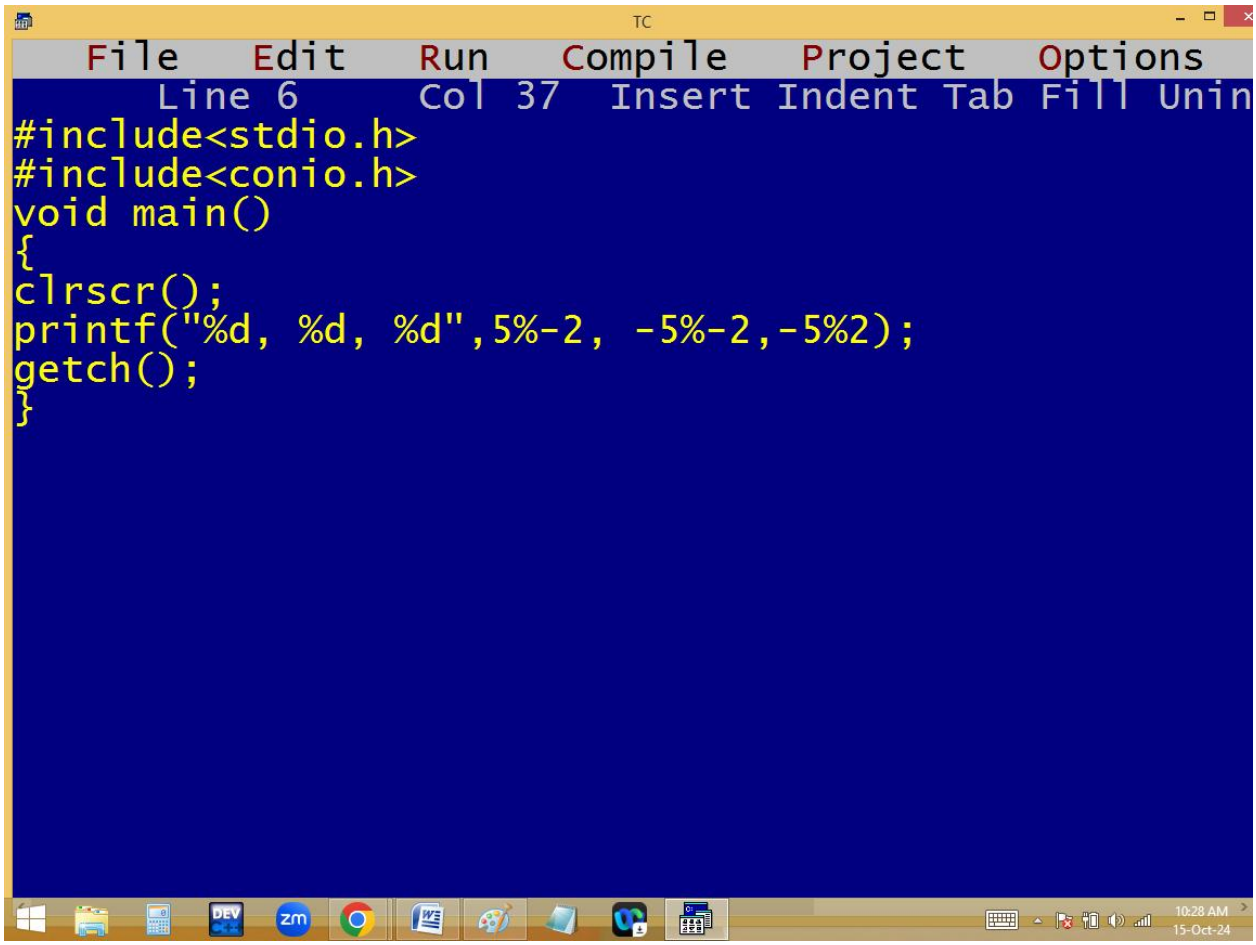


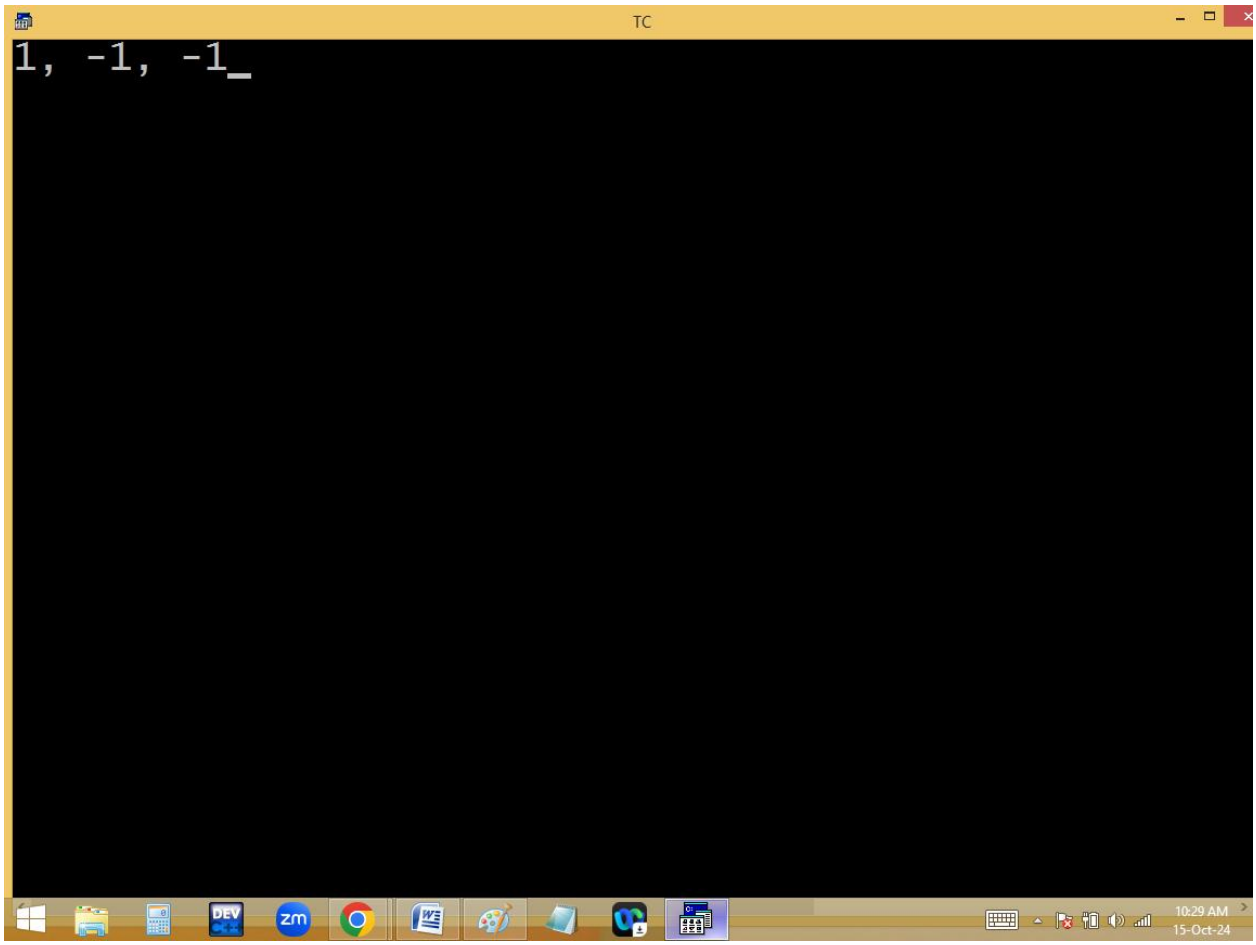
5%-2= 1

-5%-2= -1

-5%2= -1

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


TC
File Edit Run Compile Project Options
Line 6 Col 37 Insert Indent Tab Fill Unin
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d, %d, %d",5%-2, -5%-2,-5%2);
getch();
}
10:28 AM
15-Oct-24



/ - division [Quotient]:

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

5.0/2=2.500000

5/2.0=2.500000

5.0/2.0=2.500000

Note: In division both operands are int then result also int. any one or both are floats then result also float.

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

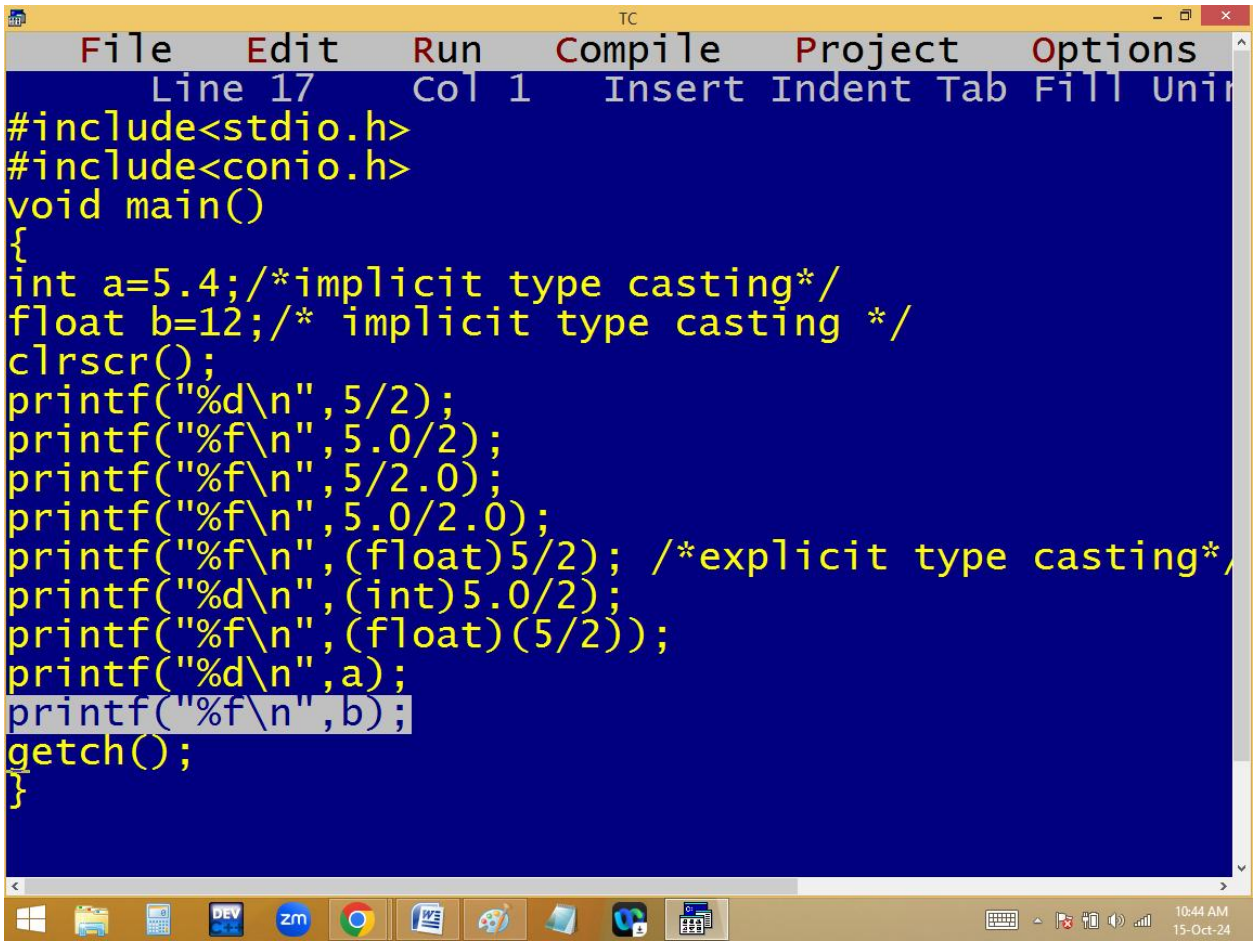
(int)5.0/2=2 [explicit]

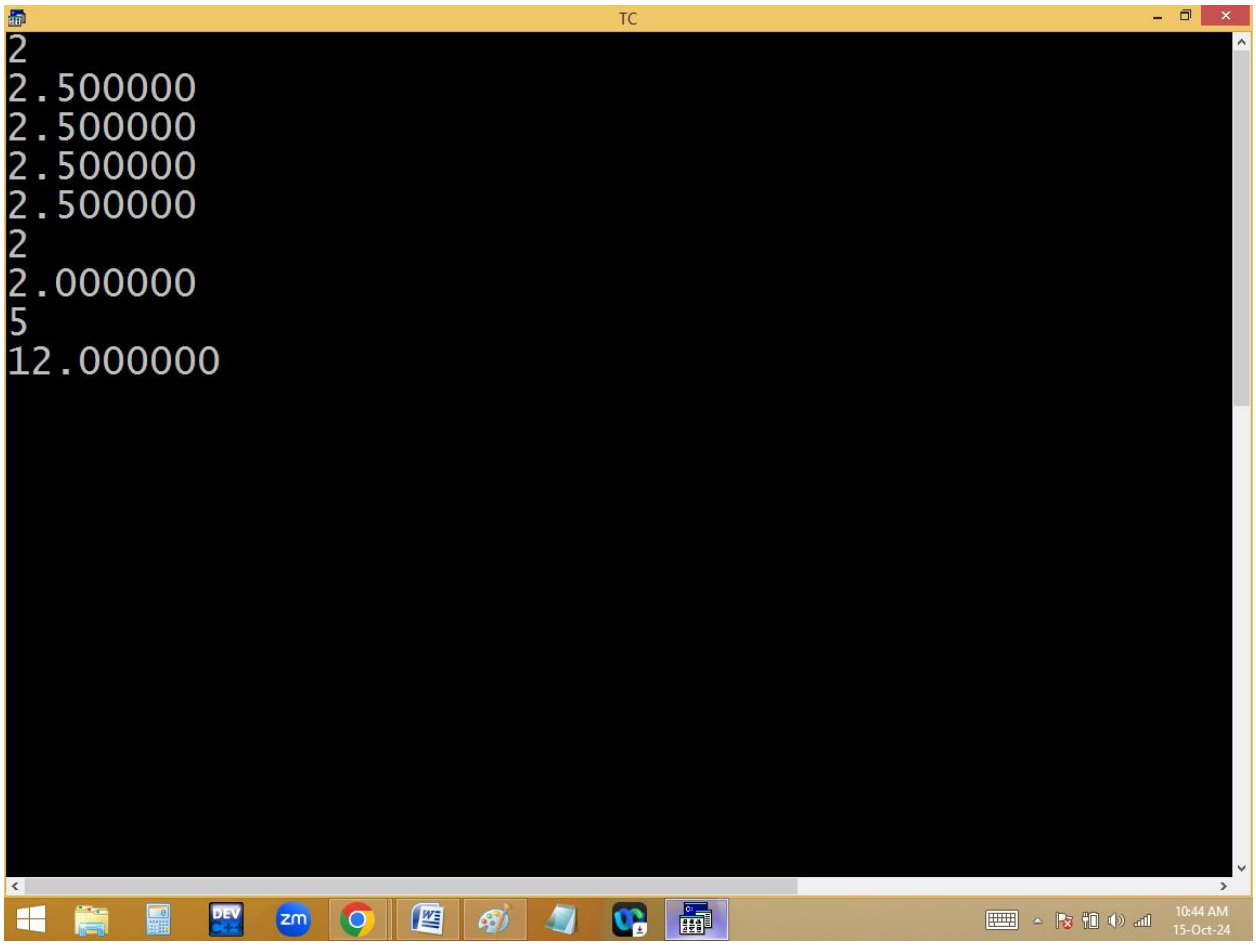
Int a = 5.4; → answer is a=5 [implicit]

Float b=12;answer is b=12.000000 [implicit]

(float)(5/2)=2.000000

(float) 5 / (int) 2.0 = 2.500000


TC
File Edit Run Compile Project Options
Line 17 Col 1 Insert Indent Tab Fill Uni
#include<stdio.h>
#include<conio.h>
void main()
{
int a=5.4;/*implicit type casting*/
float b=12;/* implicit type casting */
clrscr();
printf("%d\n",5/2);
printf("%f\n",5.0/2);
printf("%f\n",5/2.0);
printf("%f\n",5.0/2.0);
printf("%f\n",(float)5/2); /*explicit type casting*/
printf("%d\n",(int)5.0/2);
printf("%f\n",(float)(5/2));
printf("%d\n",a);
printf("%f\n",b);
getch();
}
10:44 AM
15-Oct-24


TC
2
2.500000
2.500000
2.500000
2.500000
2
2.000000
5
12.000000
10:44 AM
15-Oct-24