

## String:

A group of characters is called string.

Eg: `char city[] = "Hyd-1";`

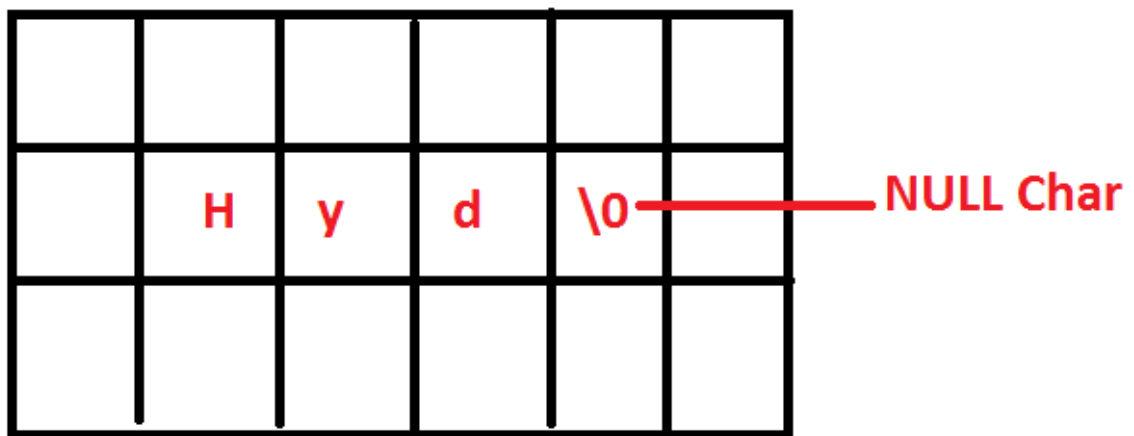
It is alpha-numeric. i.e. in a string we can store alphabets, numbers and special characters.

It is a one dimensional character array.

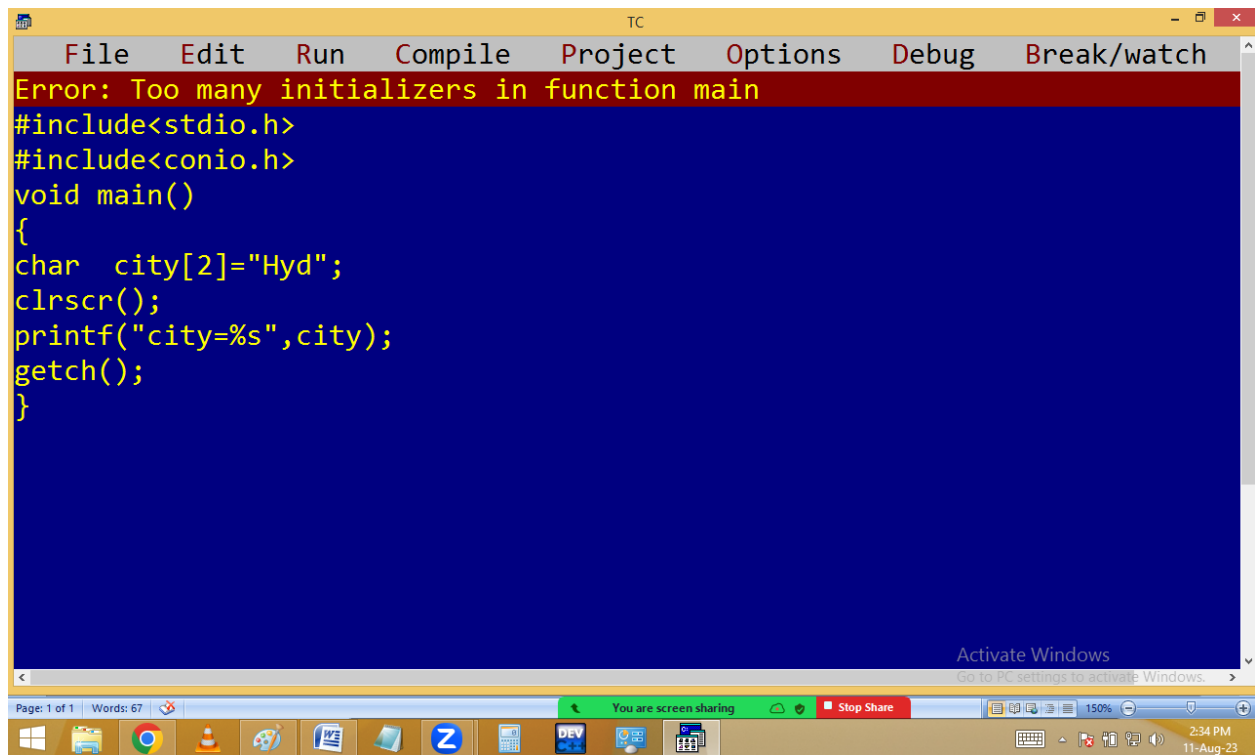
## Note:

1. One byte should be left for null char [ `'\0'` ]. Otherwise we are getting garbage values.

### RAM-BYTES



2. The string variable size never smaller than the string. Otherwise we are getting error.



The screenshot shows the Turbo C++ (TC) IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. A red error message banner at the top reads "Error: Too many initializers in function main". The code editor contains the following C code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
char  city[2]="Hyd";
clrscr();
printf("city=%s",city);
getch();
}
```

The Windows taskbar at the bottom shows various application icons, a system clock indicating 2:34 PM on 11-Aug-23, and a notification area with "You are screen sharing" and "Stop Share" buttons.

3. We can't copy a string with = operator. We have to use strcpy().

TC

File Edit Run Compile Project Options Debug Break/watch

Error: Lvalue required in function main

```
#include<stdio.h>
#include<conio.h>
void main()
{
char city[20];
city="Hyd";
clrscr();
printf("city=%s",city);
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

You are screen sharing Stop Share

TC

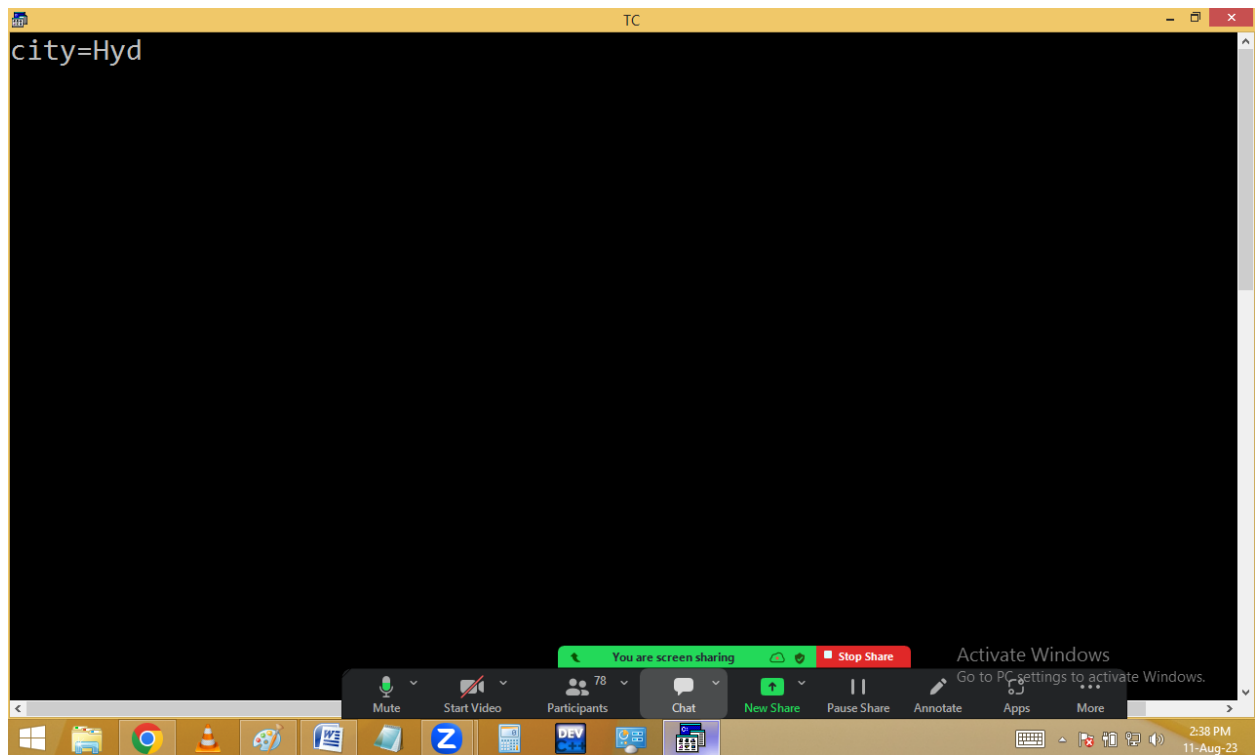
File Edit Run Compile Project Options Debug Break/watch

Line 9 Col 1 Insert Indent Tab Fill Unindent \* E:NONAME.C

```
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
char city[20];
strcpy(city,"Hyd");
clrscr();
printf("city=%s",city);
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

You are screen sharing Stop Share



4. We can't compare two strings using == [comparison] operator.

The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 25 Insert Indent Tab Fill Unindent * E:\NONAME.C
#include<stdio.h>
#include<conio.h>
#include<string.h>
void main()
{
clrscr();
if(10==10)printf("Equal\n");else printf("Not Equal\n");
if(10.5==10.5)printf("Equal\n");else printf("Not Equal\n");
if('a'=='a')printf("Equal\n");else printf("Not Equal\n");
if("ab"=="ab")printf("Equal\n");else printf("Not Equal\n");
if(strcmp("ab","ab")==0)printf("Equal\n");else printf("Not Equal\n");
getch();
}
```

The bottom window shows the output of the program:

```
Equal
Equal
Equal
Not Equal
Equal
```

Both windows have a taskbar at the bottom with various application icons and a system tray showing the time as 2:44 PM on 11-Aug-23. A green banner across the middle of the windows reads "You are screen sharing" with a "Stop Share" button.

## OPERATOR

Operator is a special symbol designed for a particular task[work]. C comes with 44 operators and 14 separators. 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, sizeof(a), +a, -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.  
**Conditional part** ? **true part** : **false part** ;

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 **left side operand should be a variable**. i.e. left side constants or expressions not allowed.

**Eg:**

a=10;

b=1.2;

c='X';

d="abc"; ➔ Error because of abc is a string

10=20; ➔ error because of 10 is constant

a = 10+20;

10+20=30; ➔ Error

a=b=c=100;

The screenshot displays the Turbo C++ (TC) IDE interface. The top window, titled 'E:NONAME.C', contains the following C code:

```
Line 12 Col 34 Insert Indent Tab Fill Unindent * E:NONAME.C
#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();
}
```

The bottom window shows the output of the program: `a=10, b=1.200000, c=X`. The Windows taskbar at the bottom indicates the time is 3:09 PM on 11-Aug-23. A green banner across the middle of the IDE windows states 'You are screen sharing' with a 'Stop Share' button. 'Activate Windows' watermarks are visible in the bottom right of both IDE windows.

TC

File Edit Run Compile Project Options Debug Break/watch

Error: Lvalue required in function main

```
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100];
clrscr();
s="Kishore";
printf("s=%s",s);
getch();
}
```

You are screen sharing Stop Share

Activate Windows Go to PC settings to activate Windows.

Mute Start Video Participants Chat New Share Pause Share Annotate Apps More

3:10 PM 11-Aug-23

TC

File Edit Run Compile Project Options Debug Break/watch

Line 7 Col 18 Insert Indent Tab Fill Unindent \* E:NONAME.C

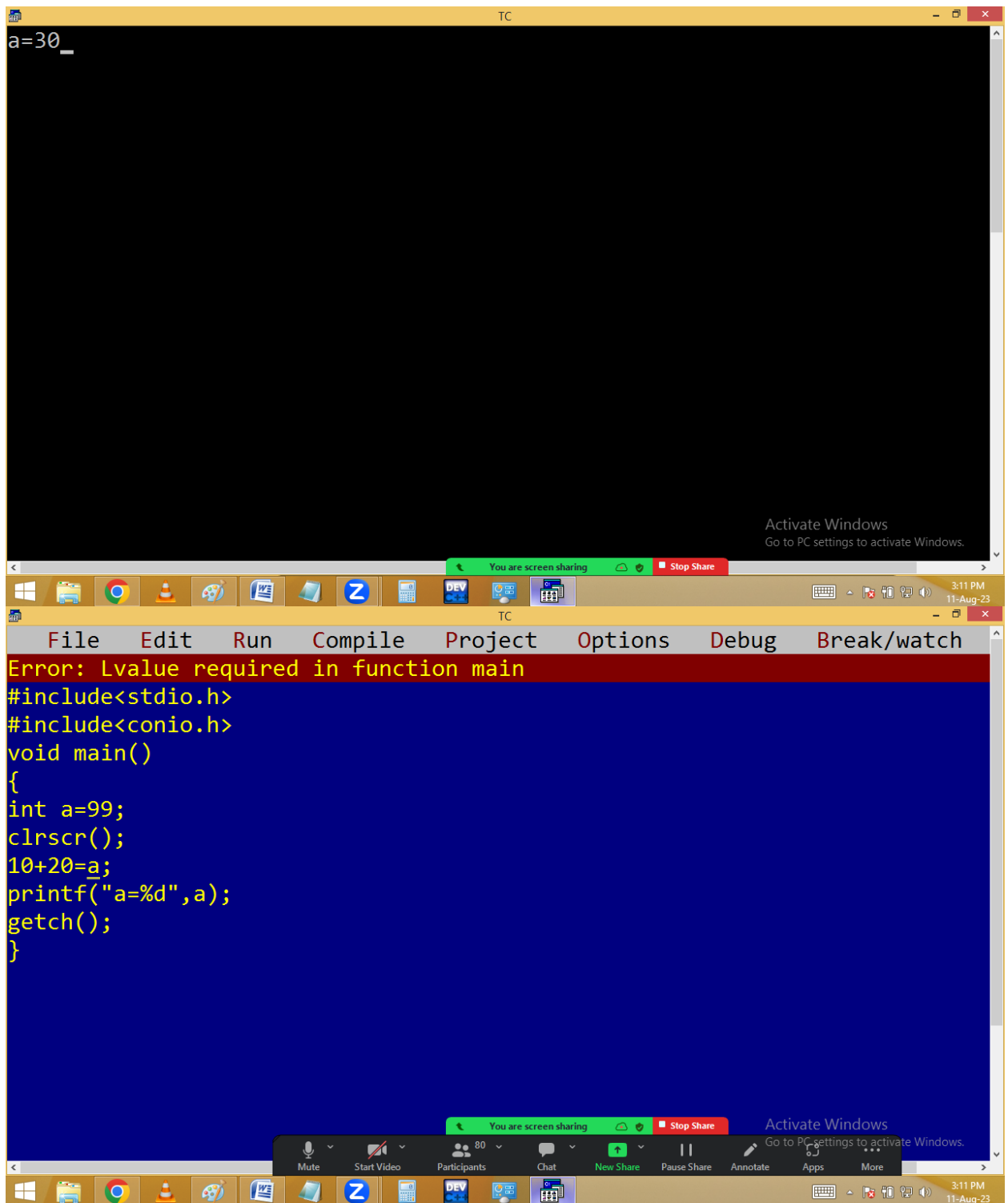
```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10+20;
clrscr();
printf("a=%d",a);
getch();
}
```

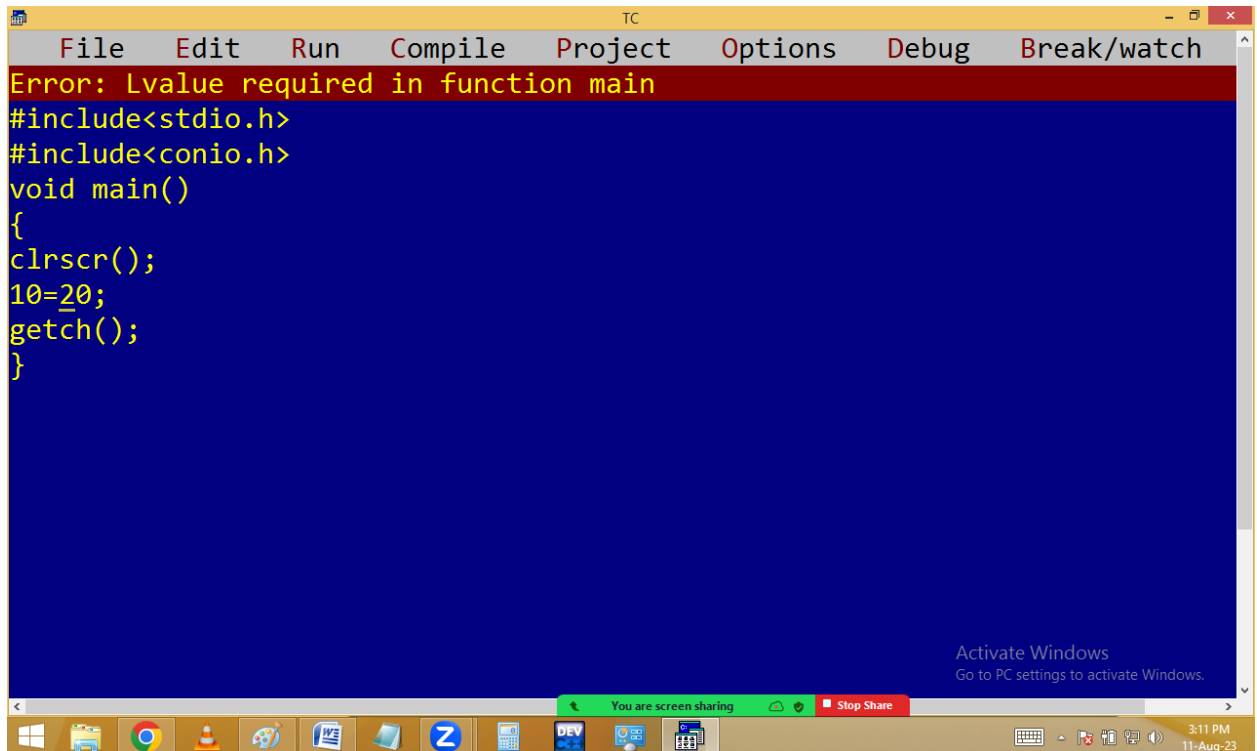
Activate Windows Go to PC settings to activate Windows.

You are screen sharing Stop Share

3:10 PM 11-Aug-23







The screenshot shows the Turbo C++ (TC) IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. A red error message banner at the top reads "Error: Lvalue required in function main". The code editor contains the following C code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
10=_20;
getch();
}
```

The Windows taskbar at the bottom shows various application icons, a system tray with a clock showing 3:11 PM on 11-Aug-23, and a notification area with "You are screen sharing" and "Stop Share" buttons.

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

Eg:  $a+b$ ,  $a-b$ ,  $a*b$ ,....

**% - Modulus [ Remainder ]:**

$5\%2=1$

$$\begin{array}{r}
 \cancel{2} \overline{) 5} \quad \cancel{2} \leftarrow \text{Quotient} \leftarrow / \\
 \underline{4} \\
 1 \leftarrow \text{Remainder}
 \end{array}$$

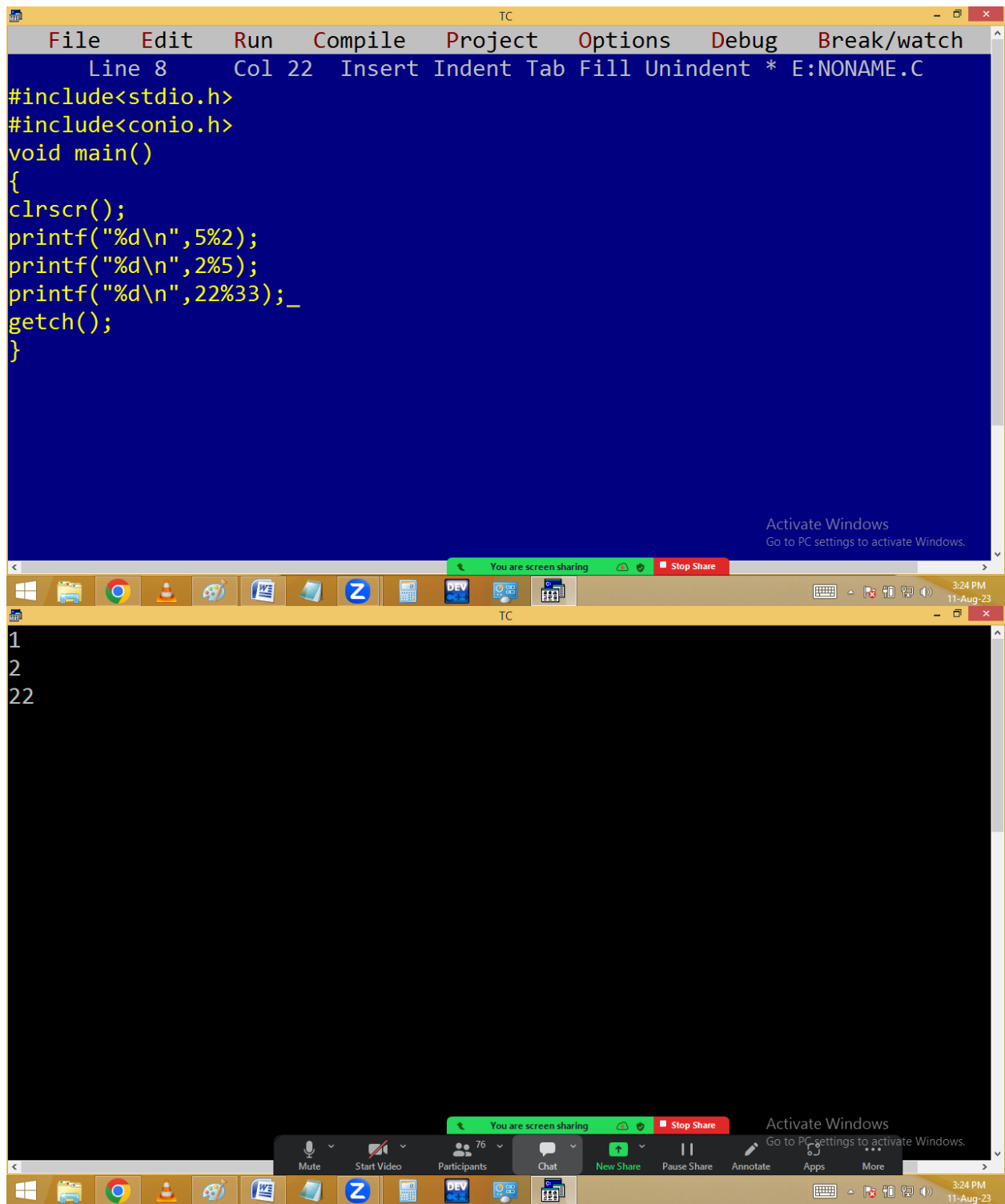
$\% \Rightarrow$

$$2\%5=2$$

**Note:** if the divisor is bigger than dividend then dividend is the result.

$$5.0 \% 3.0 = \text{Error}$$

Note: In C & C++ we can't conduct floating modulus 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("%f",5.0%2.0);
getch();
}
```

You are screen sharing Stop Share

Activate Windows Go to PC settings to activate Windows.

Mute Start Video Participants 75 Chat New Share Pause Share Annotate Apps More

3:25 PM 11-Aug-23

TC

File Edit Run Compile Project Options Debug Break/watch

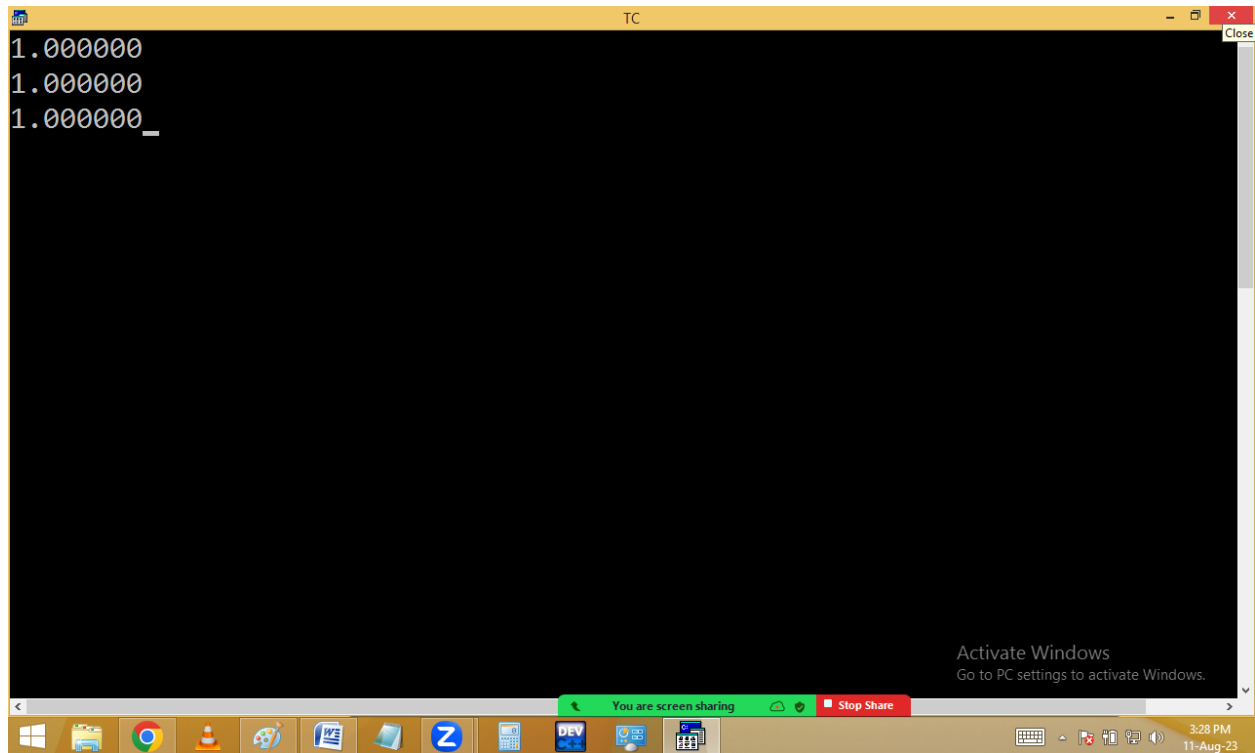
Line 8 Col 13 Insert Indent Tab Fill Unindent \* E:NONAME.C

```
#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));
printf("%f",fmod(5.0,2));
getch();
}
```

Activate Windows Go to PC settings to activate Windows.

You are screen sharing Stop Share

3:27 PM 11-Aug-23

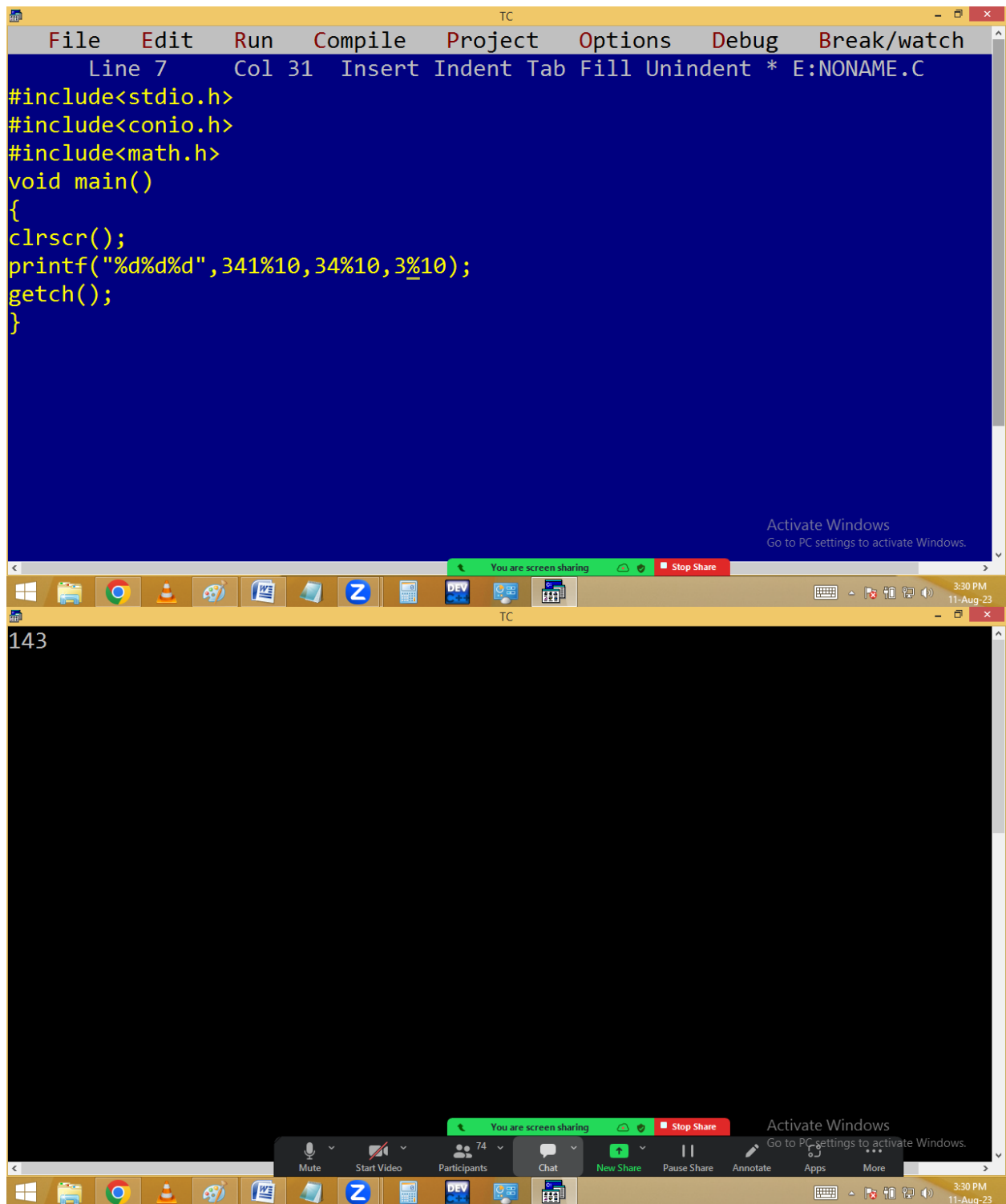


**341 % 10= 1**

**34%10=4**

**3%10=3**

**Note:** Any no%10 gives last digit.



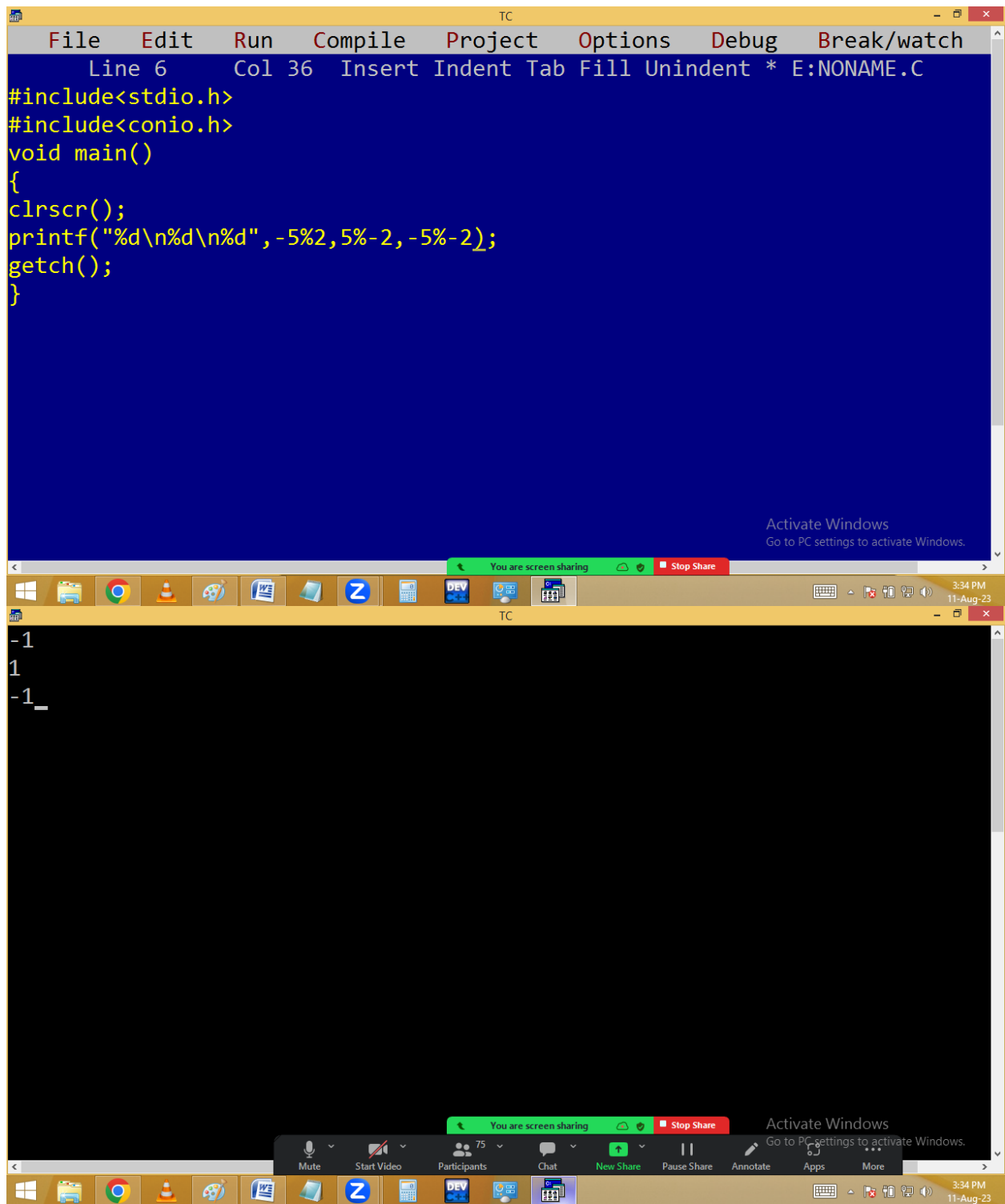
$$-5\%2 = -1$$

$$5\%-2 = 1$$

$$-5\%-2 = -1$$

**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 a=5.4; → a=5 ← implicit type casting**

**float a=5; → a=5.000000 ← implicit type casting**

**(int) 5.4/2=2 → explicit type casting**

**(float) (5/2) = 2.000000**

The image shows a Turbo C++ (TC) IDE window with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 12, Col 39, Insert, Indent, Tab, Fill, Unindent, \* E:\NONAME.C). The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=5.4; /* implicit type casting */
float b=9;
clrscr();
printf("a=%d\n",a);
printf("b=%f\n",b);
printf("%d\n",5/2);
printf("%f\n",5.0/2);
printf("%f\n",(float)5/2); /* explicit_type casting */
printf("%f\n",(float)(5/2));
printf("%d\n",(int)5.0/2);
getch();
}
```

Below the code editor, the output window displays the results of the program's execution:

```
a=5
b=9.000000
2
2.500000
2.500000
2.000000
2
```

The output window also features a status bar with options like Mute, Start Video, Participants (75), Chat, New Share, Pause Share, Annotate, Apps, and More. Both windows include a green "You are screen sharing" bar and a red "Stop Share" button. An "Activate Windows" watermark is visible in the bottom right corner of both windows.