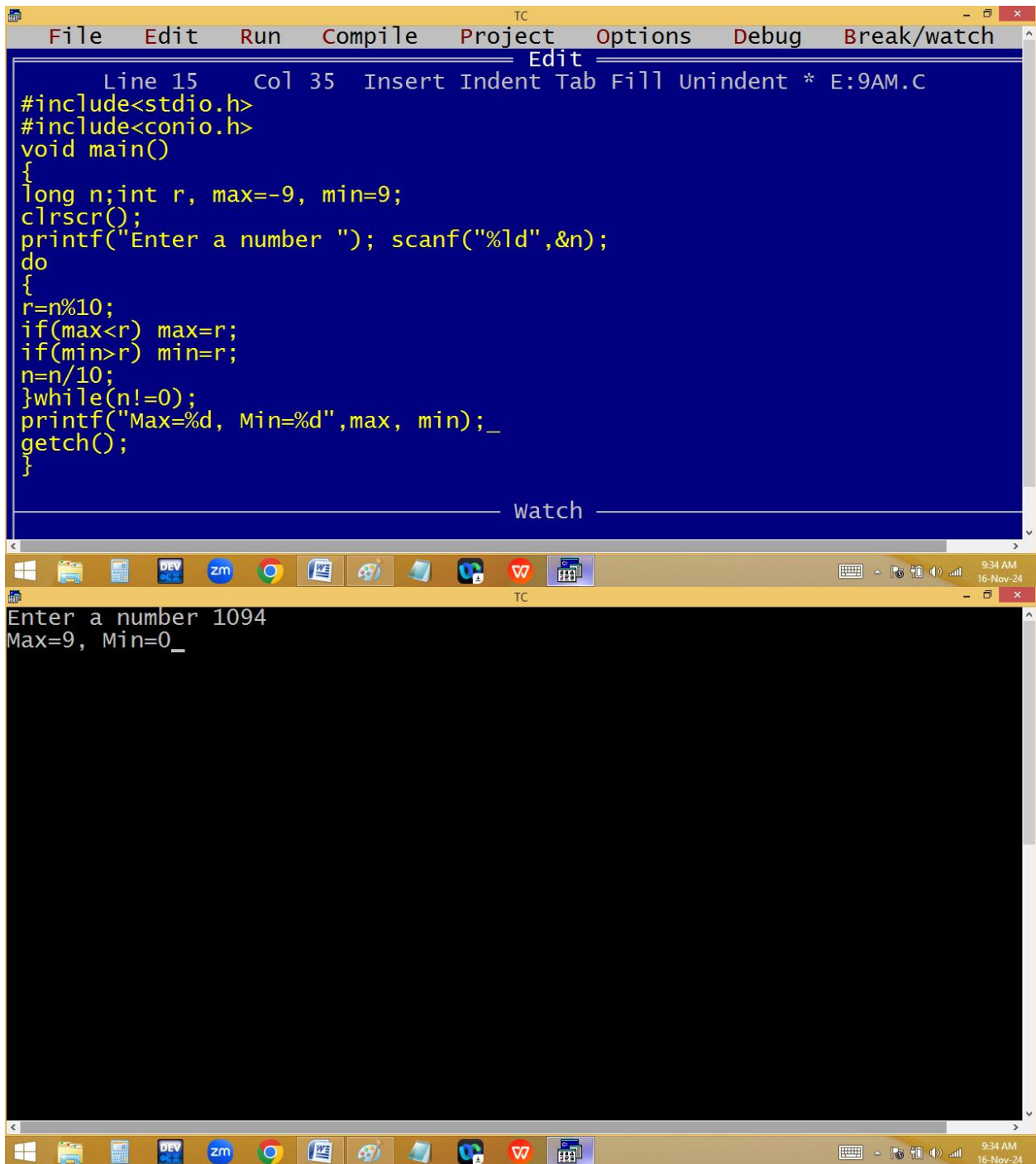


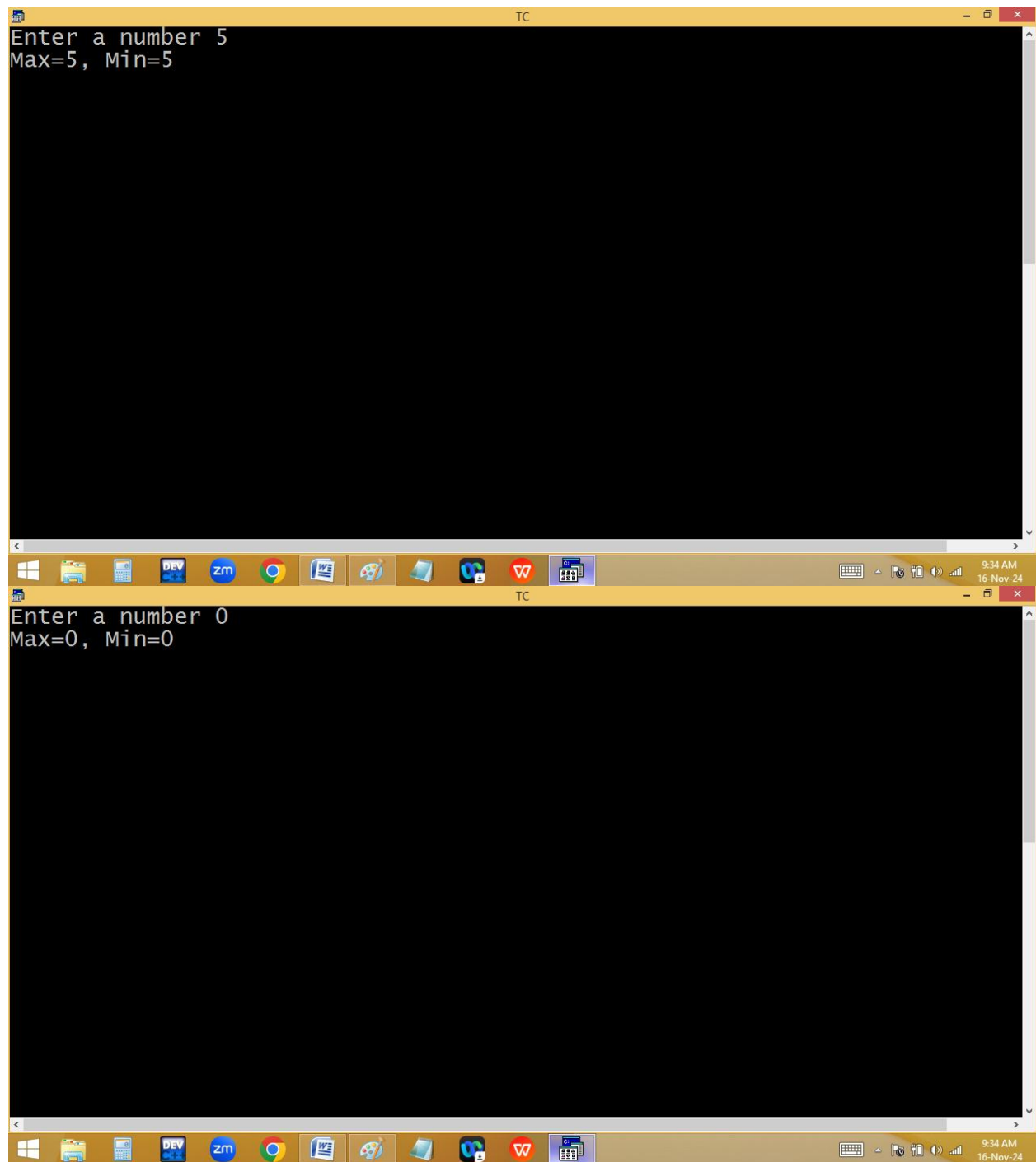
Finding max and min digits of given no:



```
TC
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 15 Col 35 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
    long n; int r, max=-9, min=9;
    clrscr();
    printf("Enter a number "); scanf("%ld",&n);
    do
    {
        r=n%10;
        if(max<r) max=r;
        if(min>r) min=r;
        n=n/10;
    }while(n!=0);
    printf("Max=%d, Min=%d",max, min);_
    getch();
}
```

Watch

Enter a number 1094
Max=9, Min=0_



```

Enter a number -1234
Max=-1, Min=-4_

```

	n	r	max	min
	1094	4	-9 < 4	9 > 4
do	104	4	4 < 9	4 > 9
{	10	0	9 < 0	4 > 0
r = n%10;	1	1	9 < 1	0 > 1
if(<u>max</u> < r) max=r; ✓				
if(<u>min</u> > r) min=r;				
n = n/10; ✓				
}while(n != 0);				
p(max, min);				

Finding reverse no:

123 reverse is 321

1 29

$$\begin{array}{l} \text{ } \rightarrow 9 \times 10 = 90 \\ \text{ } \rightarrow + 2 \end{array}$$

$$\underline{92} \times 10 = 920$$

$$\begin{array}{r} + 1 \\ \hline 921 \end{array}$$

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the editor, displaying a C program to reverse a number. The code is as follows:

```
Line 15 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
    long n, rev=0; int r;
    clrscr();
    printf("Enter a number "); scanf("%ld",&n);
    while(n!=0)
    {
        r=n%10;
        rev=rev*10+r;
        n=n/10;
    }
    printf("Reverse no is %ld",rev);
    getch();
}
```

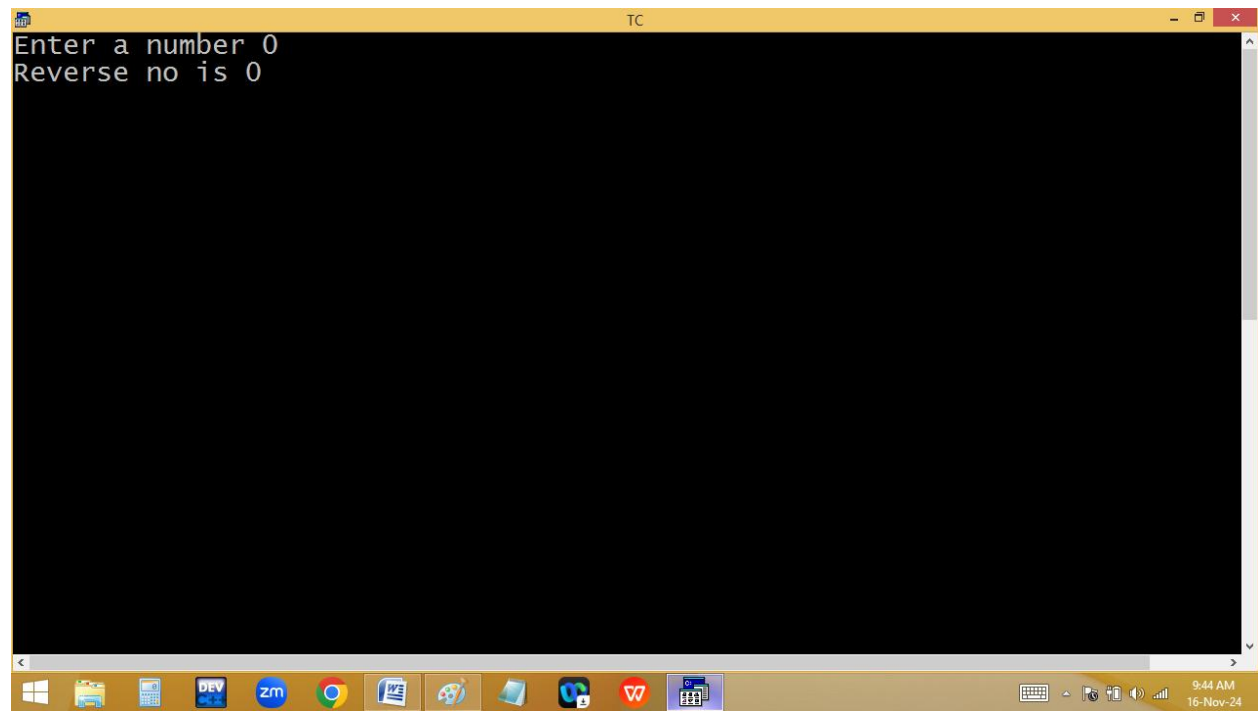
The bottom window shows the execution output of the program:

```
Enter a number 123
Reverse no is 321
```

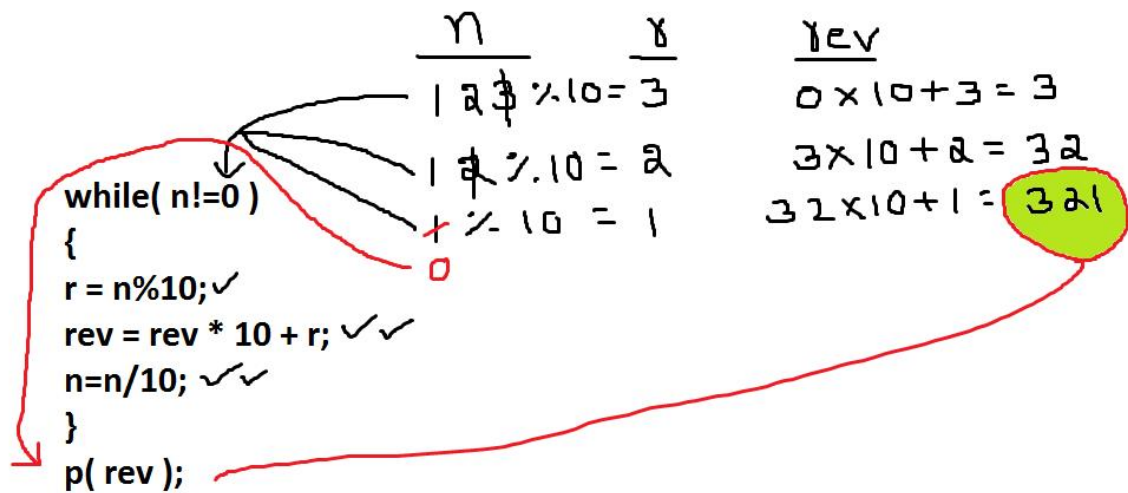
The IDE interface includes a menu bar with File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. A status bar at the bottom right shows the time as 9:43 AM on 16-Nov-24.

```
TC
Enter a number 10203456
Reverse no is 65430201_
```

```
TC
Enter a number -123
Reverse no is -321_
```



```
Enter a number 100
Reverse no is 1
```



$$\frac{n}{10} \% 10 = 0$$

$$10 \% 10 = 0$$

$$1 \% 10 = 1$$

$$\frac{rev}{10}$$

$$0 \times 10 + 0 = 0$$

$$0 \times 10 + 0 = 0$$

$$0 \times 10 + 1 = 1 \checkmark$$

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the 'Edit' window, displaying a C program to reverse a number. The code is as follows:

```
Line 8 Col 51 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
    long n;int r;
    clrscr();
    printf("Enter a number "); scanf("%ld",&n);
    printf("Reverse no is "); if(n<0)printf("-",n=-n);
    do
    {
        r=n%10;
        printf("%d",r);
        n=n/10;
    }while(n!=0);
    getch();
}
```

The bottom window is the 'Watch' window, which shows the output of the program. The user has entered the number -123, and the program has output 'Reverse no is -321_'. The taskbar at the bottom of the screen shows various application icons, including Windows Explorer, DEV C++, Zoom, Google Chrome, and others. The system clock in the bottom right corner indicates the time is 9:51 AM on 16-Nov-24.

```
TC
Enter a number 1200000
Reverse no is 0000021
```

```
TC
Enter a number 0
Reverse no is 0
```



```
TC
Enter a number -100
Reverse no is -001_
```

$$\begin{array}{r} n \\ \hline 100 \end{array} \quad \text{printf(} \underline{x} \text{)}$$

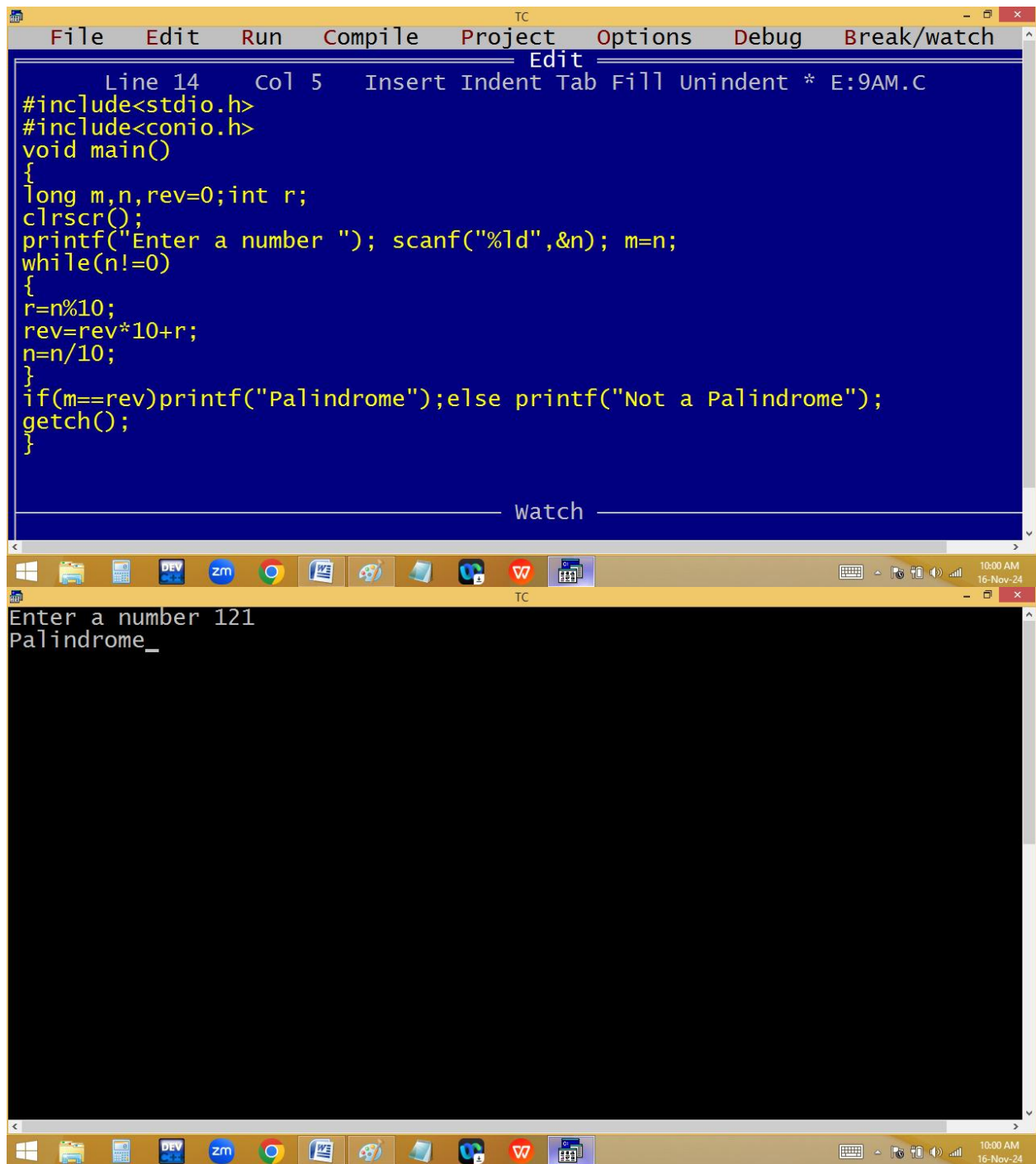
$$100 \% 10 = 0$$

$$10 \% 10 = 0$$

$$1 \% 10 = 1$$

Finding palindrome no:

121 reverse is 121



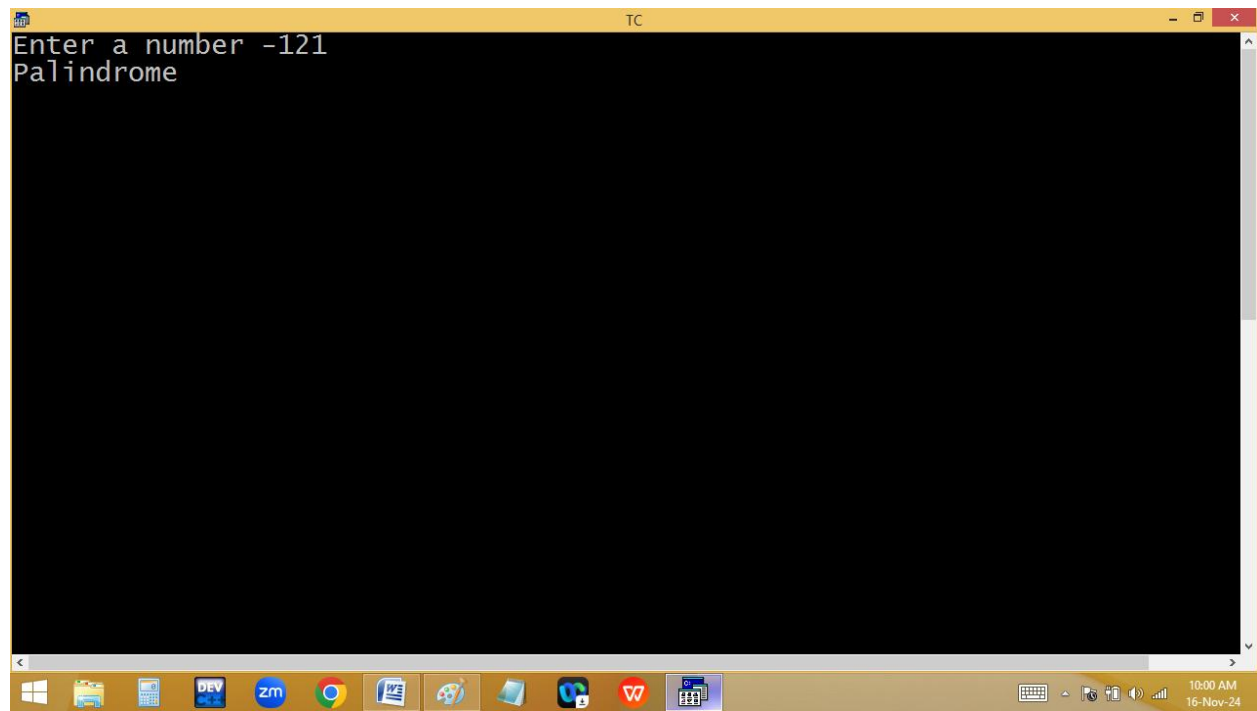
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the code editor, which has a blue background and contains the following C code:

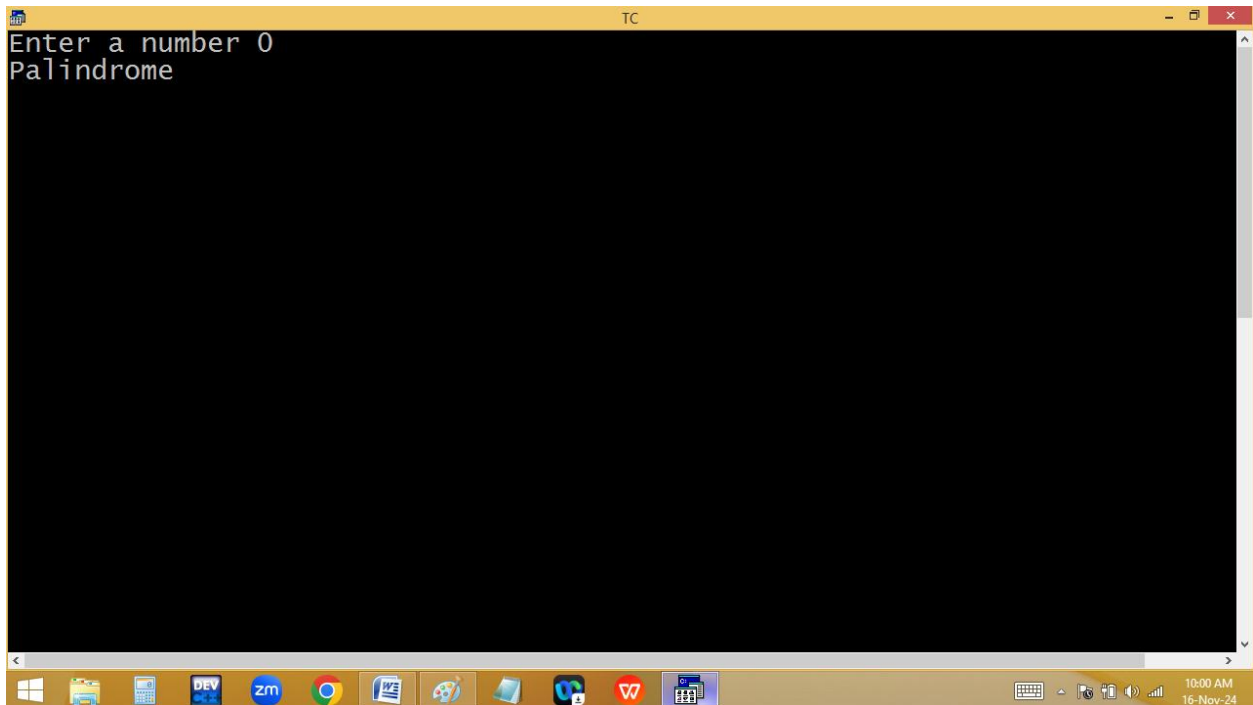
```
Line 14 Col 5 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
    long m,n,rev=0;int r;
    clrscr();
    printf("Enter a number "); scanf("%ld",&n); m=n;
    while(n!=0)
    {
        r=n%10;
        rev=rev*10+r;
        n=n/10;
    }
    if(m==rev)printf("Palindrome");else printf("Not a Palindrome");
    getch();
}
```

Below the code editor is a black console window. It displays the output of the program:

```
Enter a number 121
Palindrome_
```

The Windows taskbar is visible at the bottom of the screen, showing various application icons and the system clock indicating 10:00 AM on 16-Nov-24.





$$f \quad \frac{m}{121} = \frac{n}{121} \quad g \quad \frac{8}{121} = 1$$

$$12 \% 10 = 2$$

$$1 \% 10 = 1$$

0

$$\frac{dev}{0 \times 10 + 1 = 1}$$

$$1 * 10 + 2 = 12$$

$$12 * 10 + 1 = 121$$

No to text conversion:

102 → One Zero Two

#include<stdio.h>

#include<conio.h>


```
void main()

{

long m,n,rev=0;int r;

clrscr();

printf("Enter a number "); scanf("%ld",&n);

if(n<0) printf("-",n=-n); m=n;

while(n!=0){r=n%10;rev=rev*10+r;n=n/10;} /* reverse no */

do

{

switch(rev%10)

{

case 0: printf("Zero");break;

case 1: printf("One");break;

case 2: printf("Two");break;

case 3: printf("Three");break;

case 4: printf("Four");break;

case 5: printf("Five");break;

case 6: printf("Six");break;

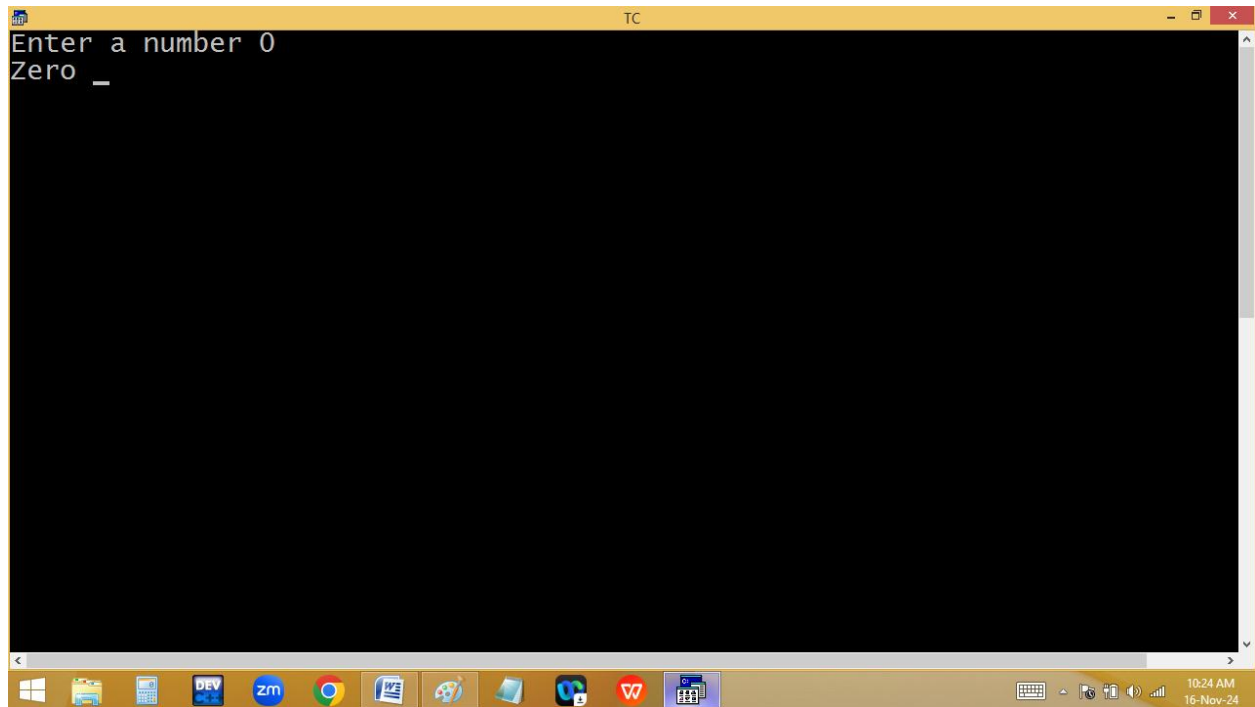
case 7: printf("Seven");break;

case 8: printf("Eight");break;
```

```
case 9: printf("Nine");break;
}
rev=rev/10;
printf(" ");
}while(rev!=0);
while(m%10==0 && m!=0)printf("Zero ",m=m/10);
getch();
}
```

```
TC
Enter a number -1230000
-One Two Three Zero Zero Zero Zero _
```

```
TC
Enter a number 102030
One Zero Two Zero Three Zero
```



Finding no of even / odd / zero digits in given no:

1023 → 1 even, 2 odd, 1 zero

```
do
{
switch( rev%10 )
{
case 0: p("Zero");b;
case 1: p("One");b;
case 2: p("Two");b;
case 9: p("Nine");
}
rev=rev/10;
}while( rev !=0);
```

$$\frac{n}{102}$$

$$\frac{rev}{207}$$

$$\begin{aligned} 207 \div 10 &= 1 \\ 20 \div 10 &= 0 \\ 2 \div 10 &= 0 \end{aligned}$$

One
Zero
Two

$$\frac{n}{1000} \div 10 = 0$$

One Zero Zero

$$\frac{n}{0}$$

while (m%10==0 && ~~m!=0~~) p("Zero", m/=10);

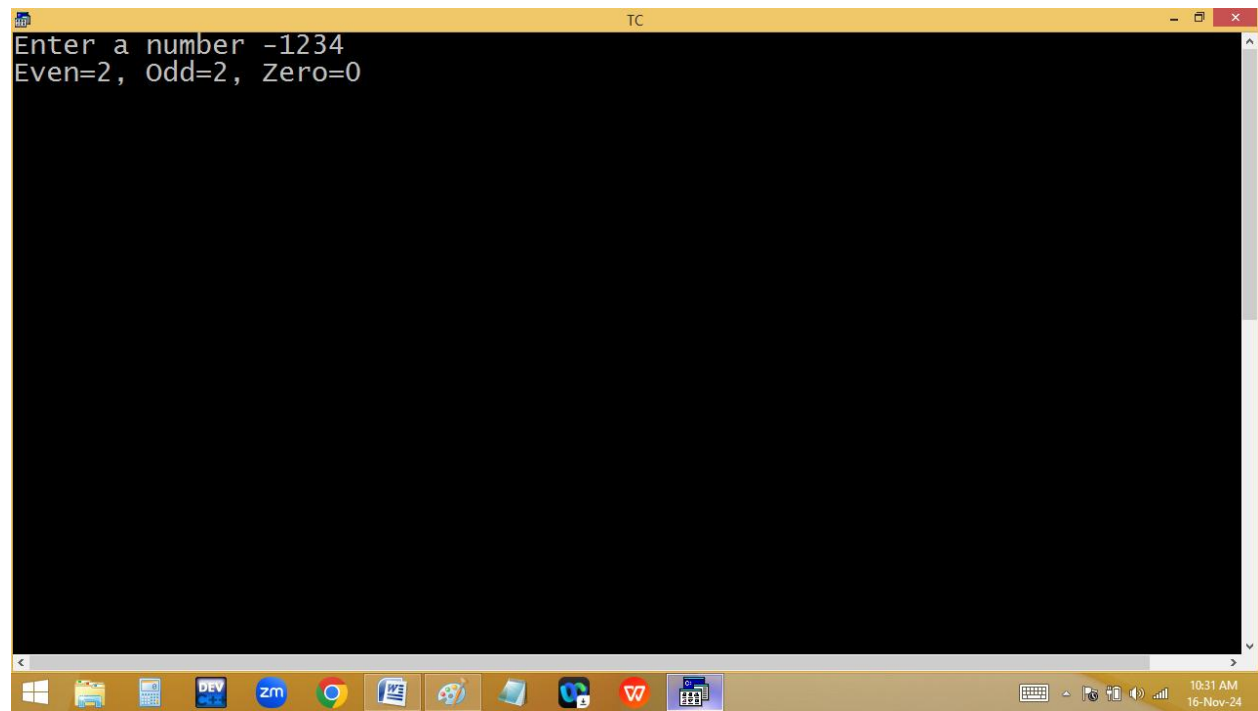
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the 'Edit' window, displaying a C program. The program prompts the user to enter a number and then counts the number of even, odd, and zero digits in that number. The code is as follows:

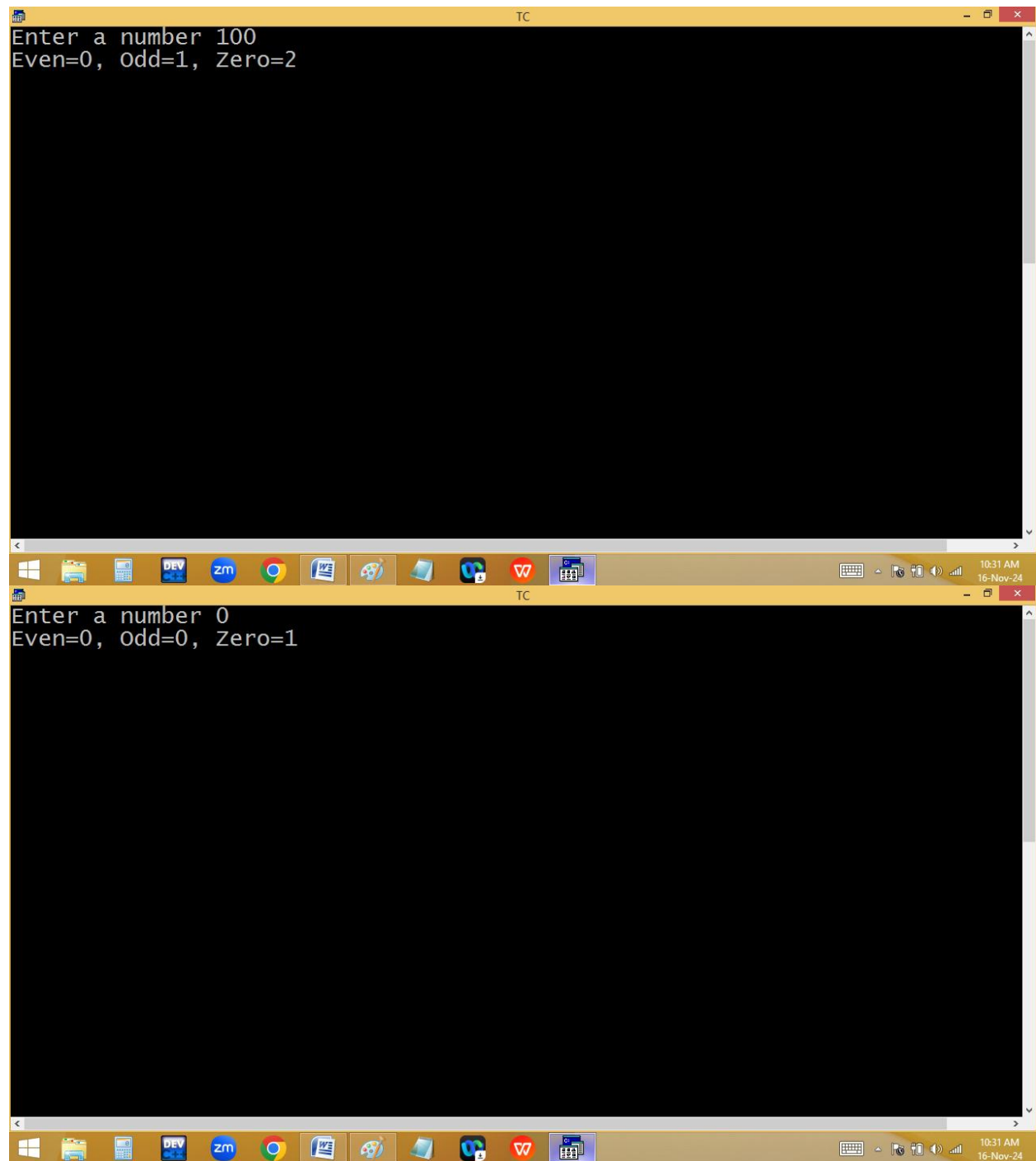
```
Line 15 Col 42 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
    long n; int r, e,o,z;
    clrscr();
    printf("Enter a number "); scanf("%ld",&n);
    e=o=z=0;
    do
    {
        r=n%10;
        if(r==0)z++;else if(r%2==0)e++;else o++;
        n=n/10;
    }while(n!=0);
    printf("Even=%d, Odd=%d, Zero=%d",e,o,z);_
    getch();
}
```

Below the 'Edit' window is the 'Watch' window, which is currently empty. At the bottom of the IDE is the 'Output' window, which shows the execution results:

```
Enter a number 1023
Even=1, Odd=2, Zero=1
```

The Windows taskbar is visible at the bottom of the screen, showing the time as 10:30 AM and 10:31 AM on 16-Nov-24.





```

do
{
r=n%10;
if(r==0)z++;
else if(r%2==0)e++;
else o++; ✓ ✓
n=n/10; ✓
}while(n!=0);
p( e, o, z );

```

n	r	e	o	z
1023	3	0	0	0

102 % 10 = 2 (1)

10 % 10 = 0 (1)

~~1~~ % 10 = 1 (2)

□

for loop:

It is an entry control loop.

for is a keyword.

It is also used to repeat a program several times based on a condition.

When compared with while and do while, for loop is looking to be smart. In for it is compulsory to maintain two semicolons. **For works without condition also and default condition is always 1 i.e. true.**

Generally for loop is having 3 expressions.

1. Initialization
2. Test condition / expression
3. Increment/decrement / updation

At first entry of for loop the initialization part is executed and later the test condition is checked. If the condition is true then the for block statements are executed. After completion of the block, the increment or decrement part is executed. Later once again the test condition is evaluated. If it is true then

once again for block statements are executed. Like this the process is continued until the condition becomes false. Here the initialization part is executed only once, at the time of loop beginning.

It is mandatory to maintain 2 semicolon (;) in a for loop.

If the for loop is having more than three expressions, it is mandatory to separate the expressions with , separator.

If the for loop is having less than three expressions, then leave the expressions with empty semicolon.

```

..;
..;
for( initialization ; condition ; incr / decr / update )
{
..;
..;
}
..;
..;

```

Diagram illustrating the execution flow of a `for` loop:

- ① points to the `initialization` part.
- ② points to the `condition` part.
- ③ points to the `incr / decr / update` part.
- A green arrow labeled "true" loops from the end of the loop body back to the `condition` part.
- A red arrow labeled "false" loops from the end of the loop body back to the `..;` statement following the loop.

```

for( ; ; )
{
}

```

```

for( exp ; exp ; exp )
{
..;
}

```

```

for( exp, exp ; exp ; exp, exp )
{
}

```

```

for( ; exp ; )
{
}

```