

## POINTERS

Pointer is a variable, which holds the address of another variable of same type.

Pointer is a memory location, which holds the address of another memory location.

Pointer is a derived data type.

### Advantages:

1. Dynamic memory allocation.
2. Program performance is increased due to preventing memory wastage.
3. They are very much used in System programming.
4. They are very much used in dynamic linked list & Stacks [**data structures**].
5. **It allows to access local variable outside the function i.e. data sharing between functions.**  
[ **call by address** ].
6. **To handle strings, arrays etc in functions we need pointers.**

7. To handle **data files** we are using pointers.
8. They directly works on variable address. Due to this search time is reduced and execution speed is increased.

### Dis-advantage:

They are not secured.

### Syntax:

**datatype \* variable;**

- \* indicates it is a pointer data type.
- \* is called indirection operator.
- \* is called dereferencing operator.
- **\* is a re-direction operator.**
- \* indicates value at that address.
- \* indicates pointer value.

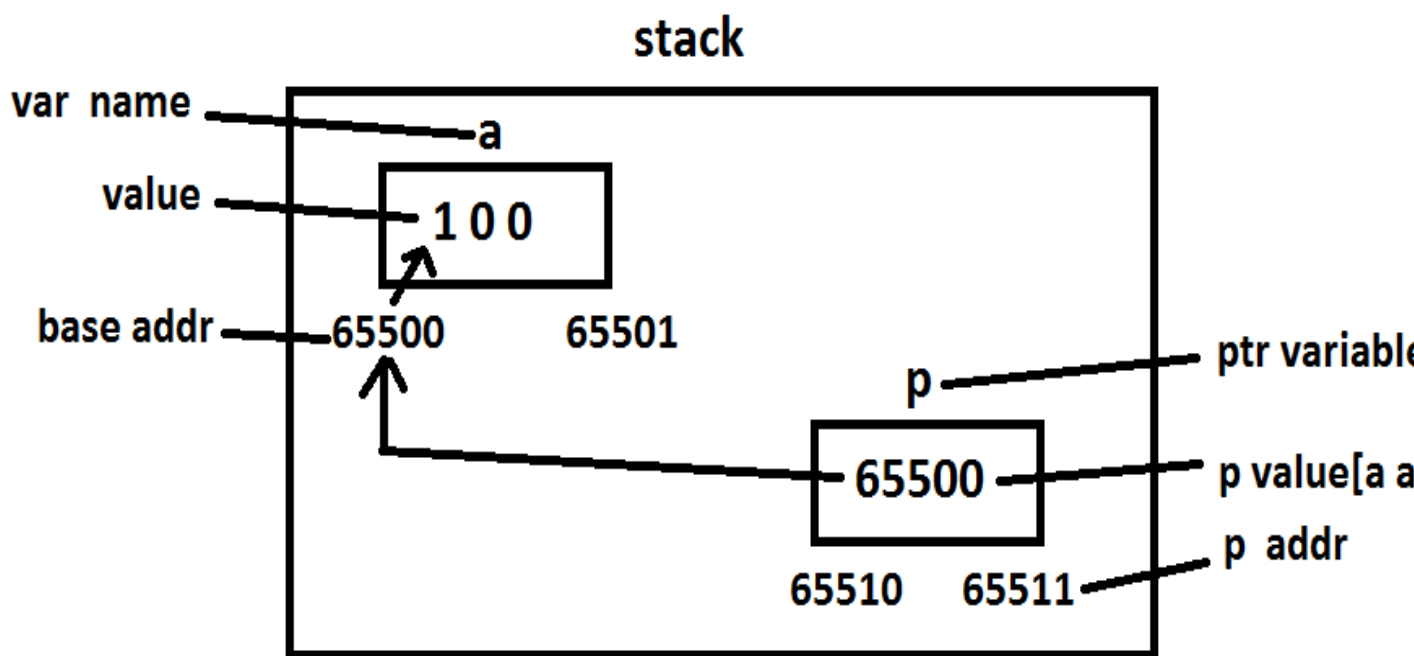
**Eg:**

```
int a=100, * p;
```

In the above example '**a**' is a general variable.

\* indicates '**p**' is a pointer type variable and it is able to store the address of general variable '**a**' as follows.

```
p = &a;
```



In the above example, to pick the value of **a** through pointer variable **p**, we have to use the **printf()** as follows.

```
printf( "%d", *p );
```

→ **100**

Here **\*p** means **value of p** or **value at that addr.** i.e. **65500**. But **65500** is the **addr of 'a'**. The **value in a address is 100**.

**Or**

Here **p** means **65500**. **\*p** means **value at 65500**. i.e. **100**.

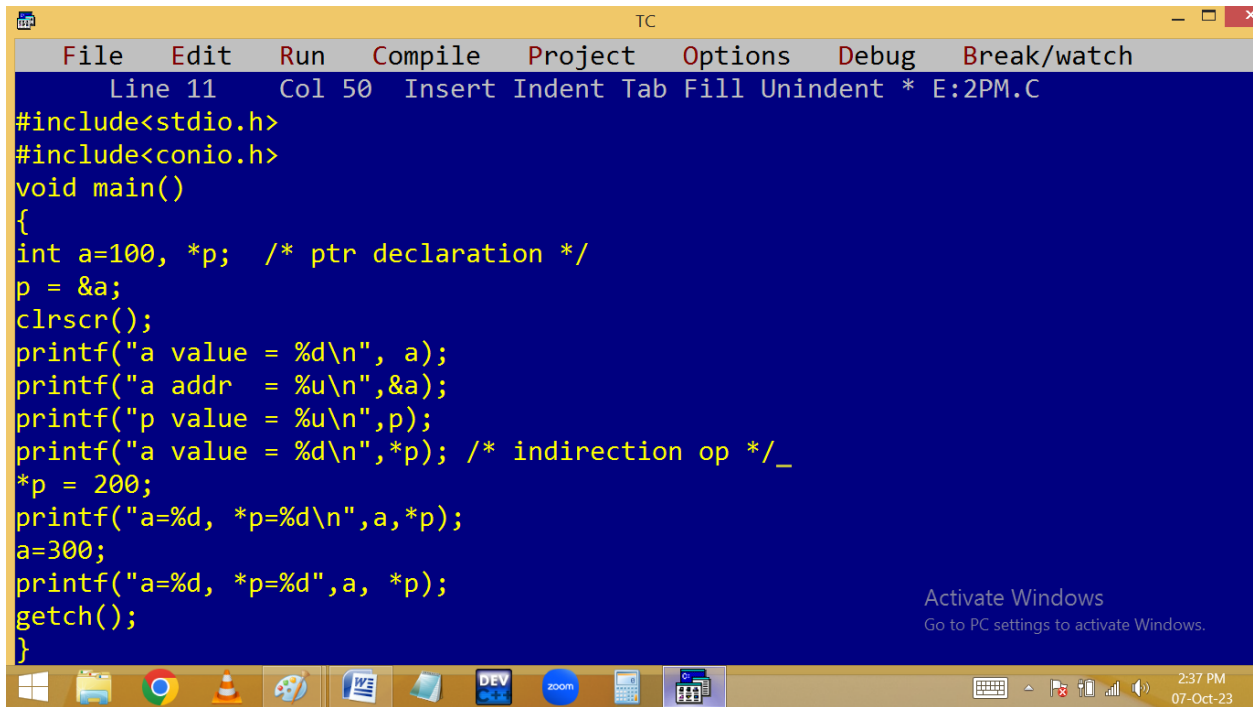
Due to this example any changes conducted in **\*p** effects the value of **'a'**. Hence **p** is called **pointer to a**.

Eg: **\*p=200;**

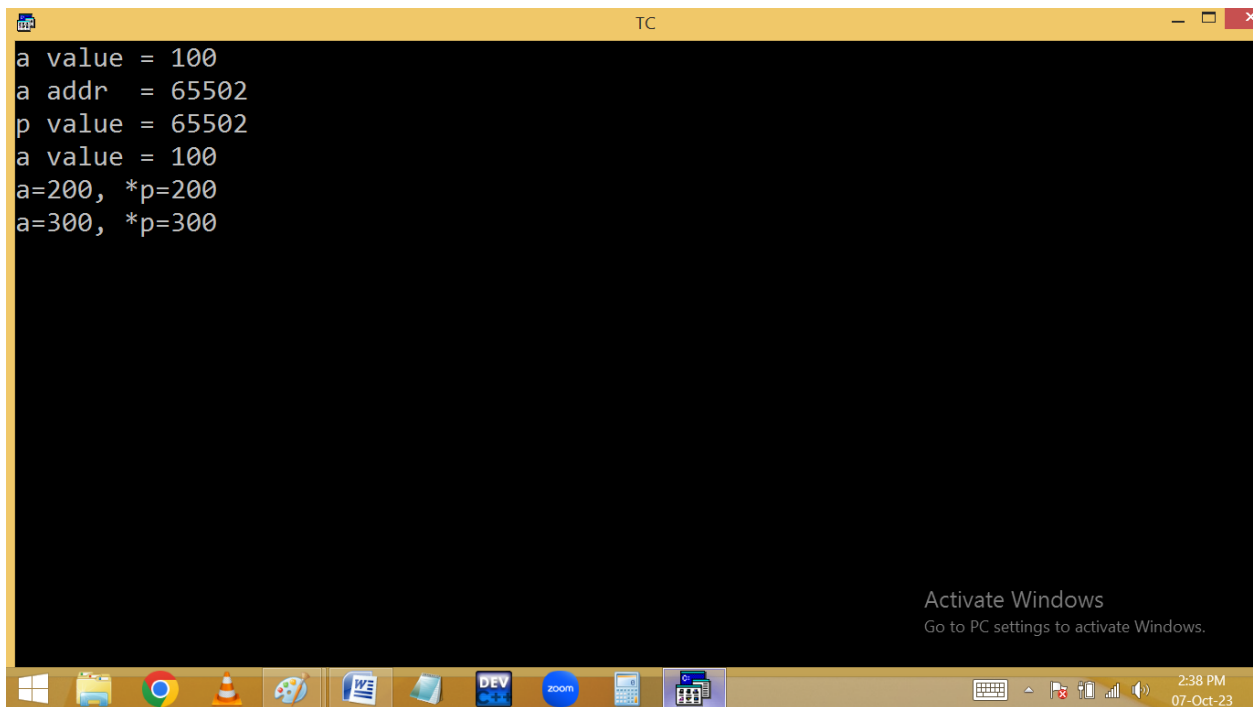
Now **a** value becomes **200**.

**Eg:**

**Finding a variable value and address using a pointer:**

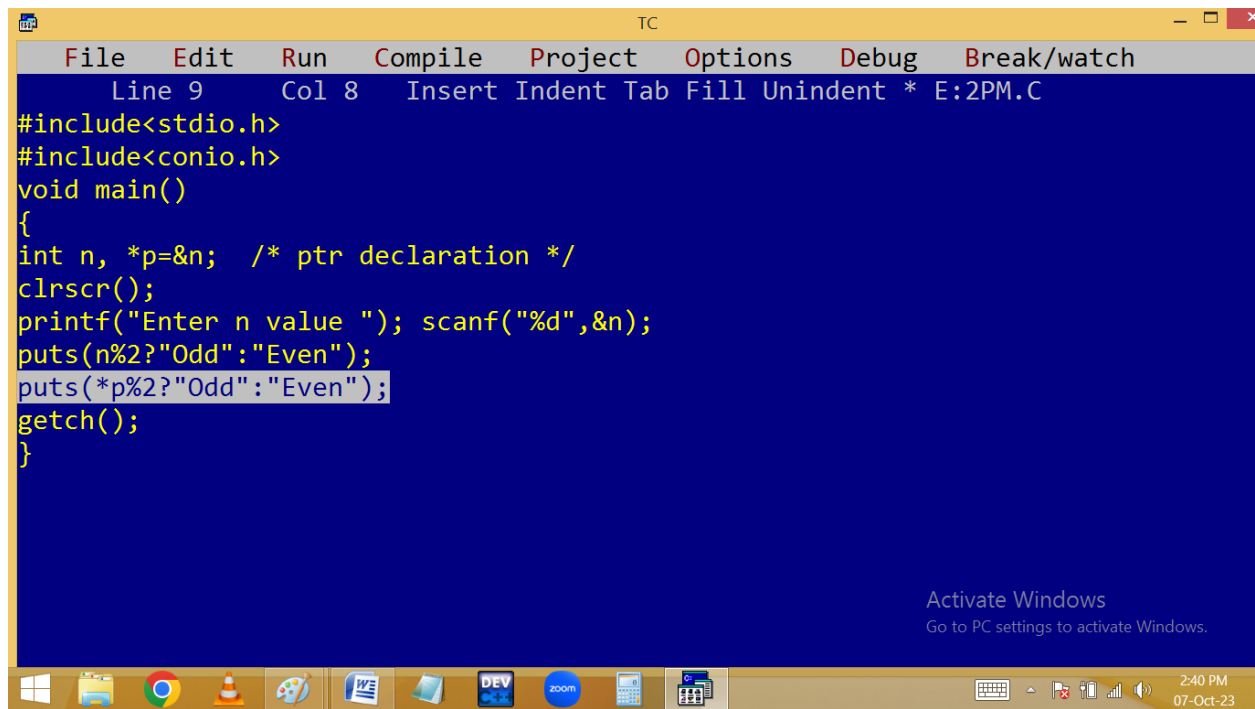


```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 50 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=100, *p; /* ptr declaration */
p = &a;
clrscr();
printf("a value = %d\n", a);
printf("a addr  = %u\n",&a);
printf("p value = %u\n",p);
printf("a value = %d\n",*p); /* indirection op */
*p = 200;
printf("a=%d, *p=%d\n",a,*p);
a=300;
printf("a=%d, *p=%d",a, *p);
getch();
}
```



```
TC
a value = 100
a addr  = 65502
p value = 65502
a value = 100
a=200, *p=200
a=300, *p=300
```

**Finding even/odd using pointer:**



TC

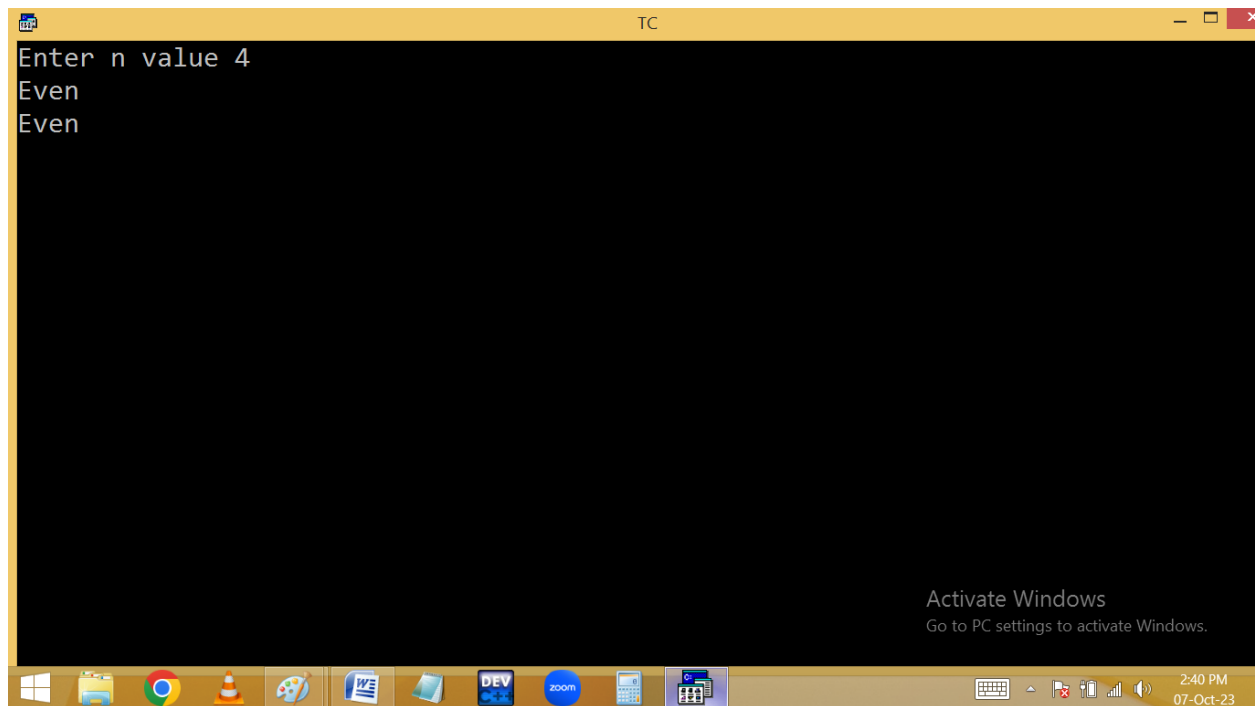
File Edit Run Compile Project Options Debug Break/watch

Line 9 Col 8 Insert Indent Tab Fill Unindent \* E:2PM.C

```
#include<stdio.h>
#include<conio.h>
void main()
{
int n, *p=&n; /* ptr declaration */
clrscr();
printf("Enter n value "); scanf("%d",&n);
puts(n%2?"Odd":"Even");
puts(*p%2?"Odd":"Even");
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

2:40 PM  
07-Oct-23

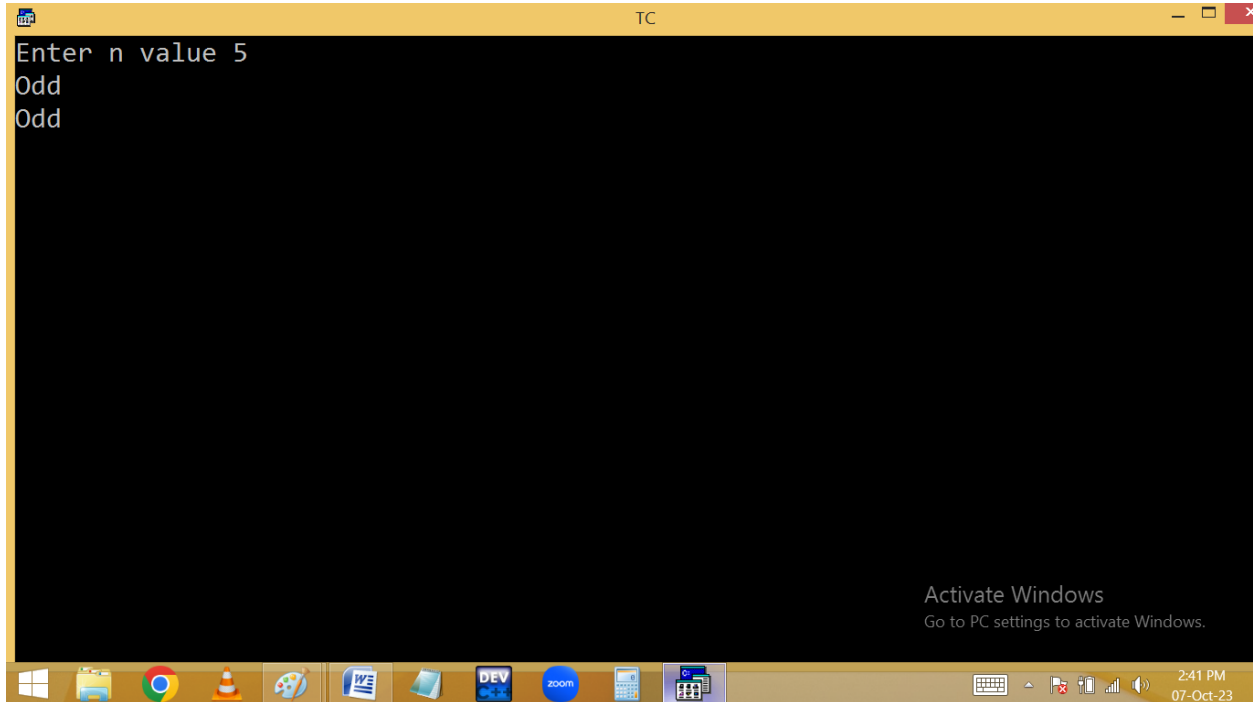


TC

Enter n value 4  
Even  
Even

Activate Windows  
Go to PC settings to activate Windows.

2:40 PM  
07-Oct-23

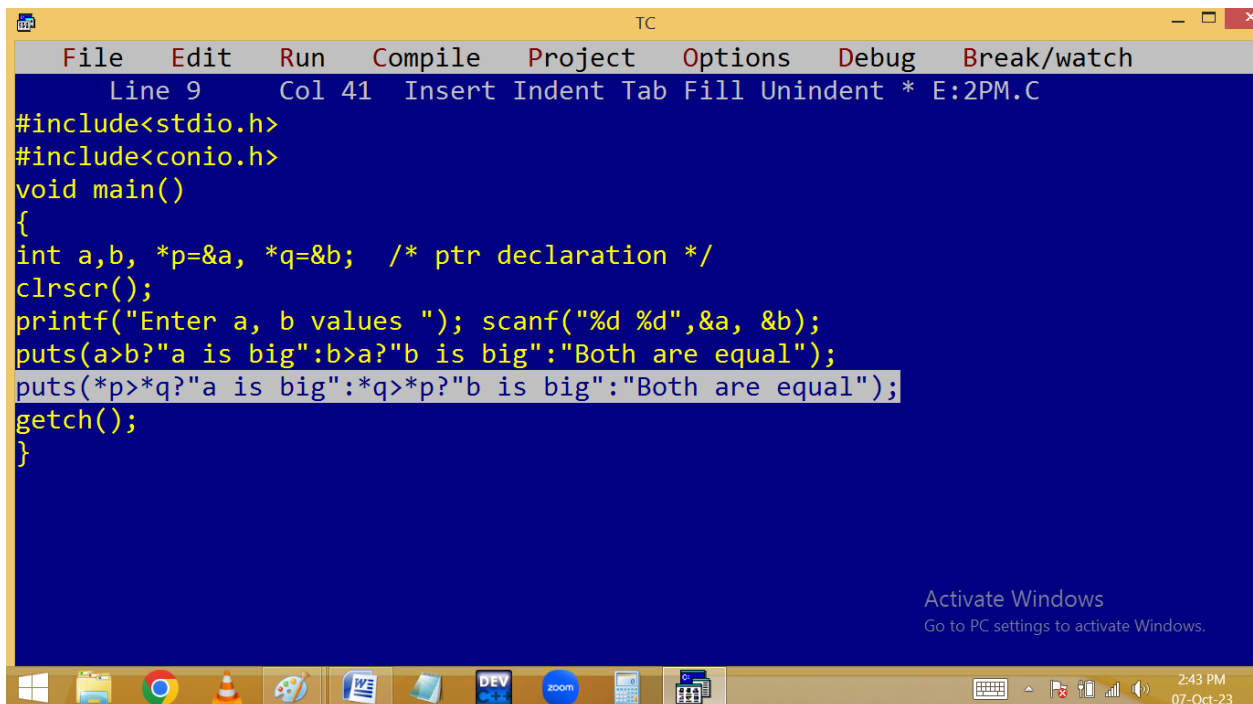


```
TC
Enter n value 5
Odd
Odd

Activate Windows
Go to PC settings to activate Windows.

2:41 PM
07-Oct-23
```

## Finding max in 2 numbers using pointers:



```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 41 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b, *p=&a, *q=&b; /* ptr declaration */
clrscr();
printf("Enter a, b values "); scanf("%d %d",&a, &b);
puts(a>b?"a is big":b>a?"b is big":"Both are equal");
puts(*p>*q?"a is big":*q>*p?"b is big":"Both are equal");
getch();
}
```

```
TC
Enter a, b values 2 3
b is big
b is big
```

Activate Windows  
Go to PC settings to activate Windows.

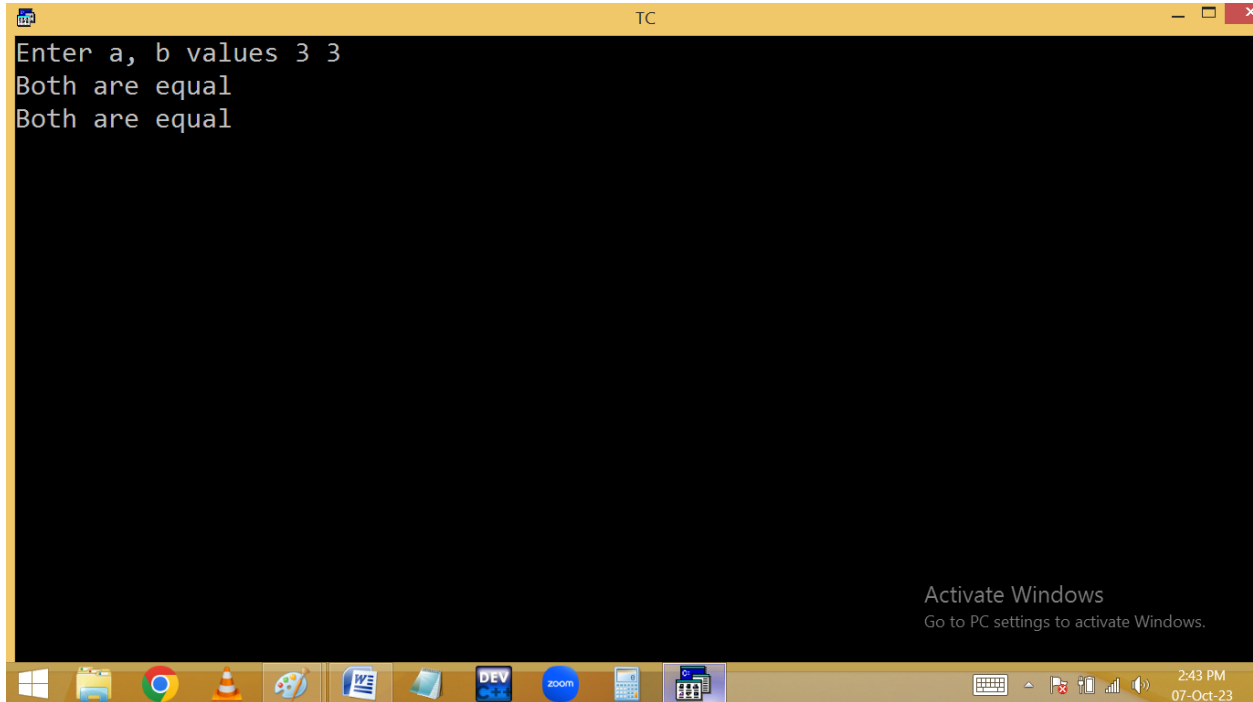
2:43 PM  
07-Oct-23

```
TC
Enter a, b values 5 1
a is big
a is big
```

Activate Windows  
Go to PC settings to activate Windows.

2:43 PM  
07-Oct-23



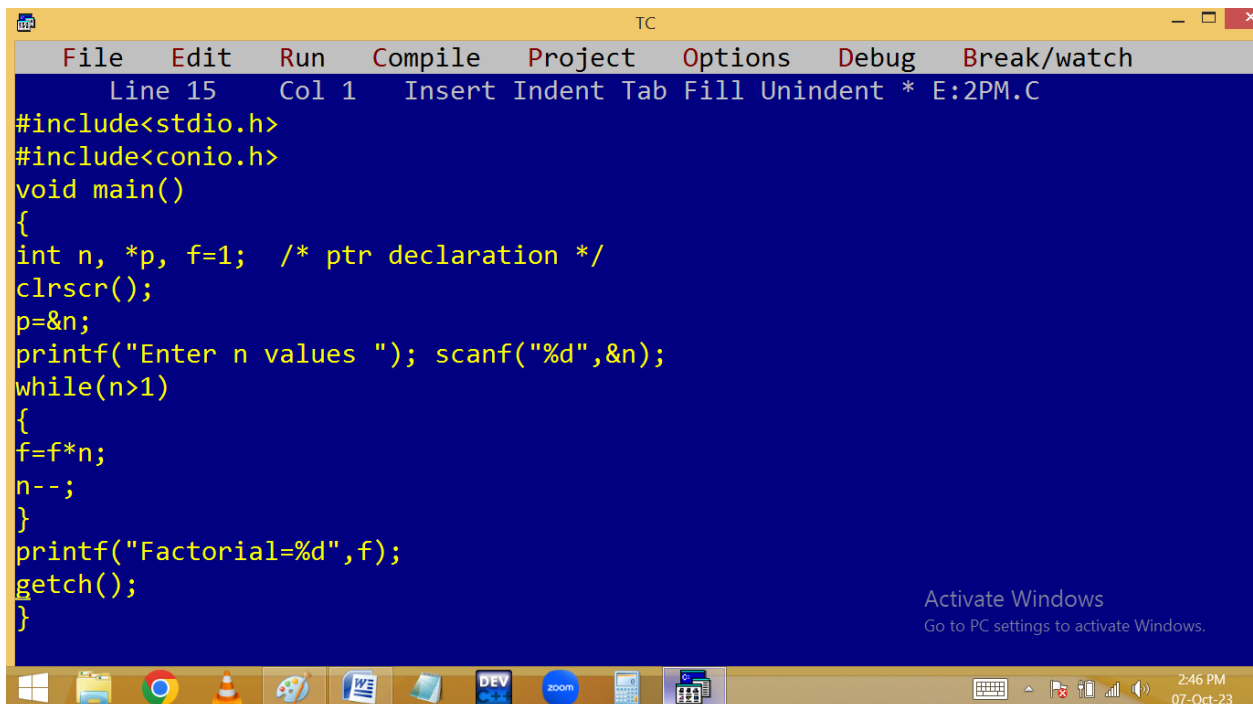


The screenshot shows a Turbo C++ (TC) console window with a black background and white text. The text displayed is:

```
Enter a, b values 3 3
Both are equal
Both are equal
```

The window title bar says "TC". At the bottom right, there is a watermark that says "Activate Windows Go to PC settings to activate Windows." The Windows taskbar is visible at the bottom with various icons and a system clock showing 2:43 PM on 07-Oct-23.

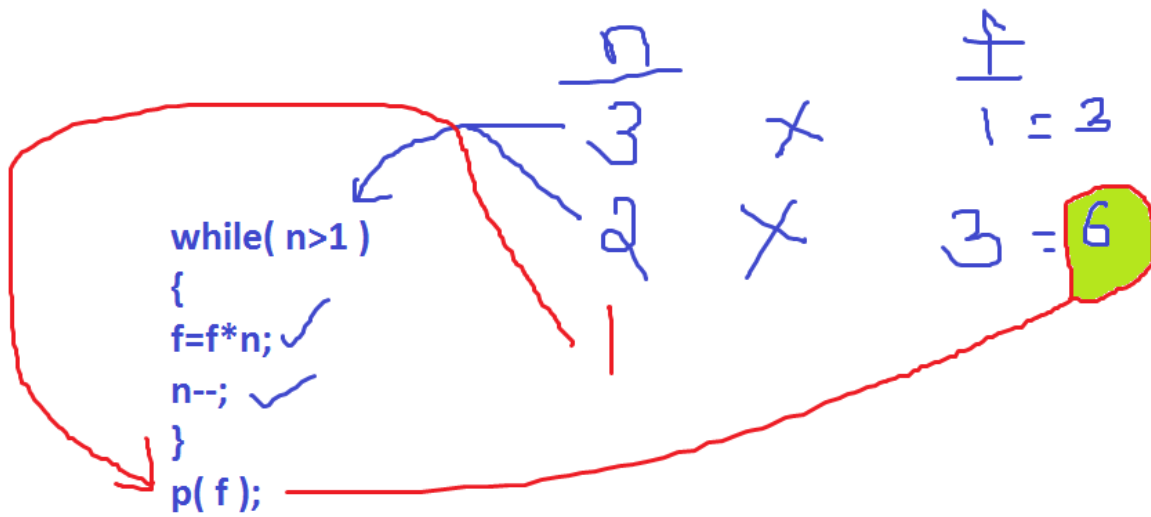
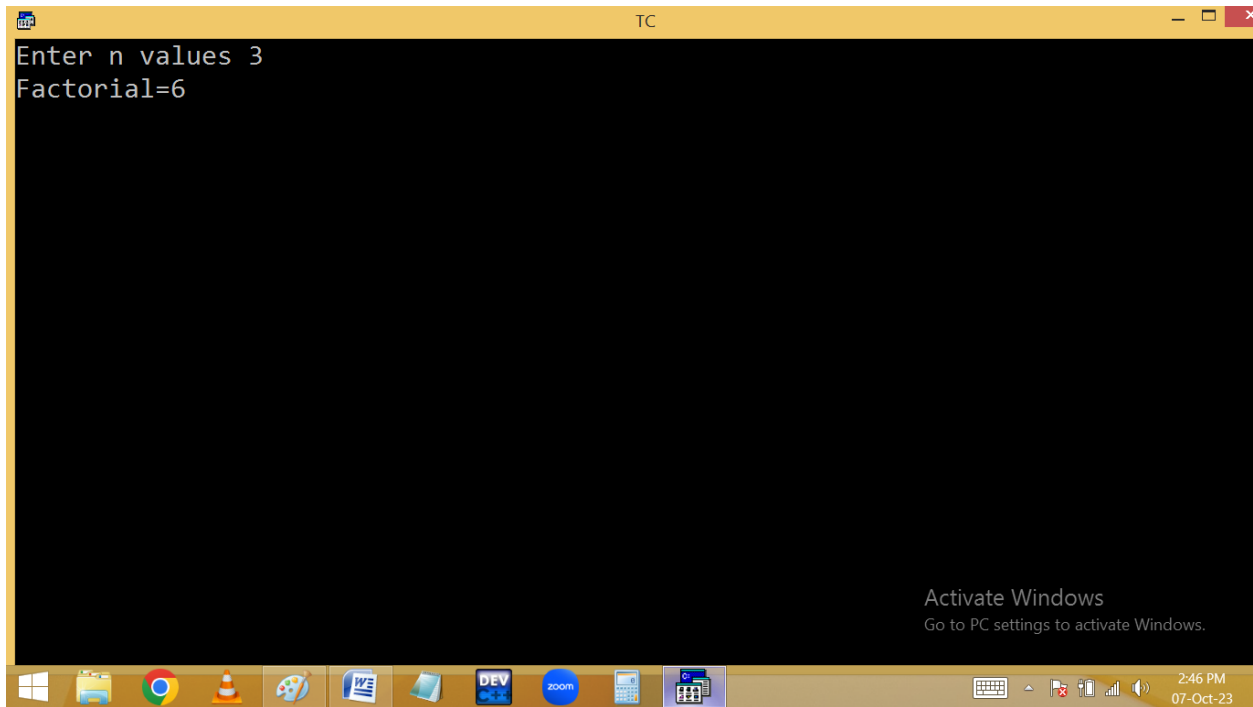
## Finding factorial using pointer:

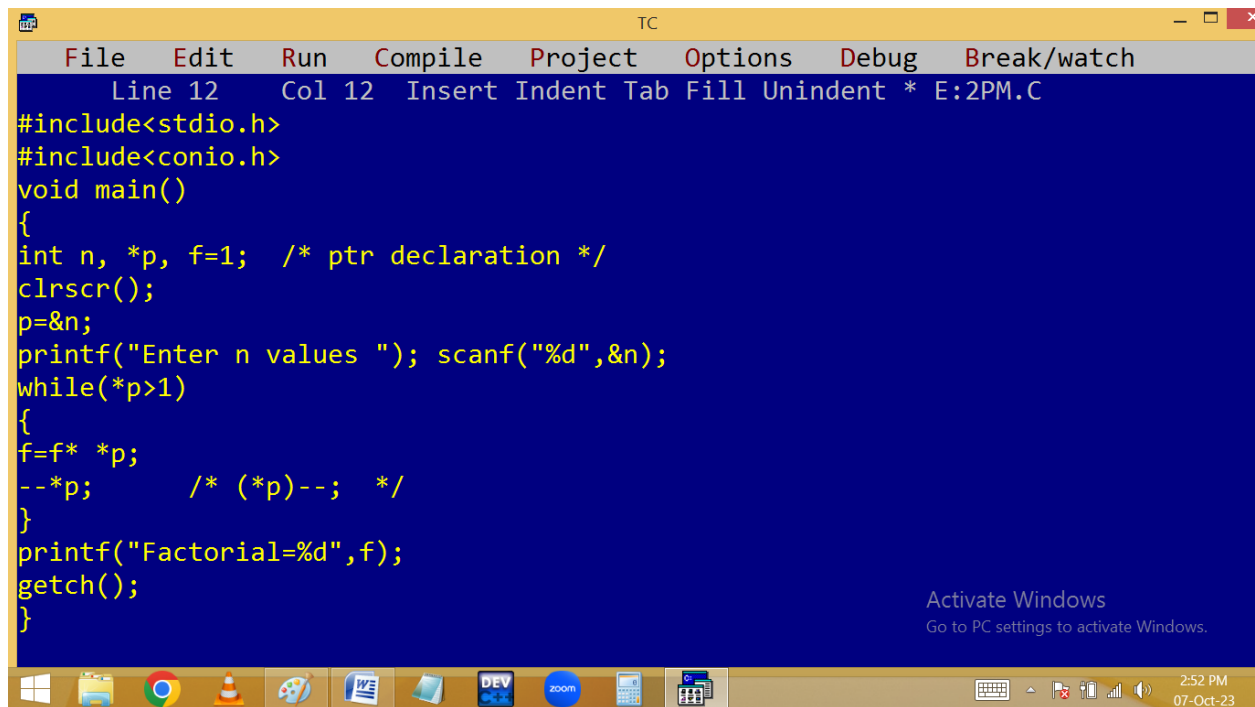


The screenshot shows a Turbo C++ (TC) editor window with a blue background and yellow text. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar shows "Line 15 Col 1 Insert Indent Tab Fill Unindent \* E:2PM.C". The code is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int n, *p, f=1; /* ptr declaration */
    clrscr();
    p=&n;
    printf("Enter n values "); scanf("%d",&n);
    while(n>1)
    {
        f=f*n;
        n--;
    }
    printf("Factorial=%d",f);
    getch();
}
```

The window title bar says "TC". At the bottom right, there is a watermark that says "Activate Windows Go to PC settings to activate Windows." The Windows taskbar is visible at the bottom with various icons and a system clock showing 2:46 PM on 07-Oct-23.

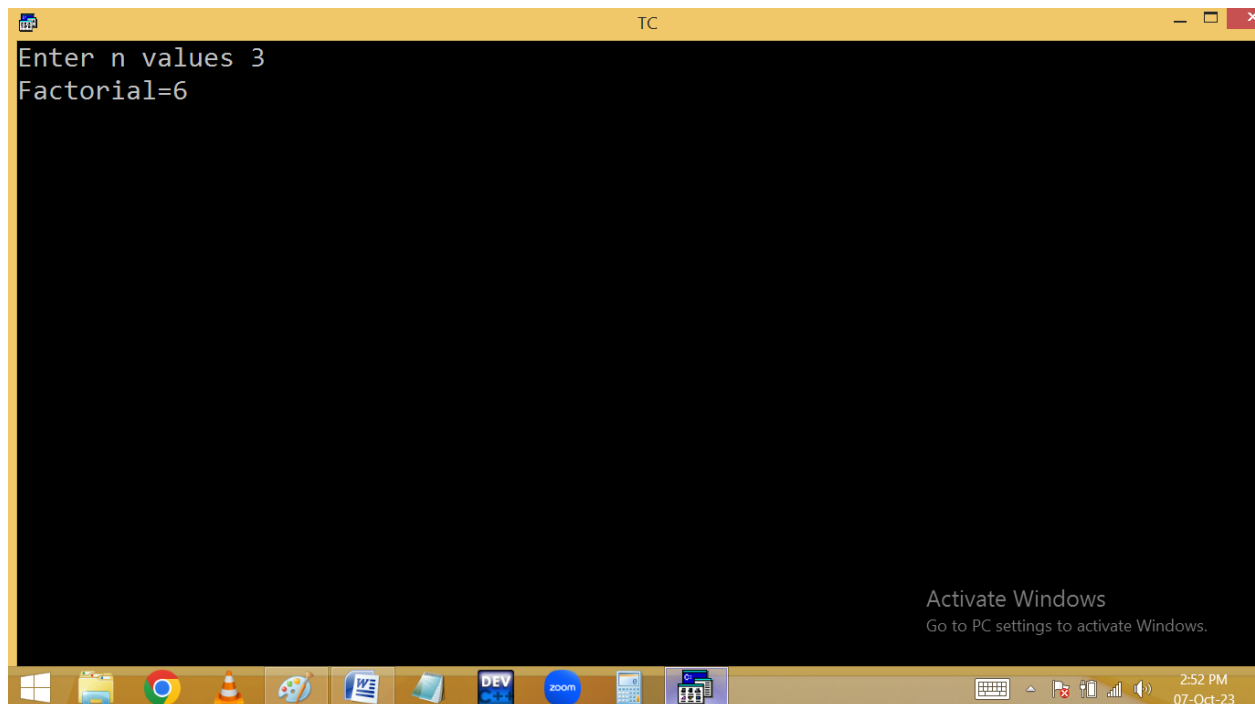




The screenshot shows the Turbo C++ (TC) IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a toolbar. The code editor displays the following C program:

```
Line 12 Col 12 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n, *p, f=1; /* ptr declaration */
clrscr();
p=&n;
printf("Enter n values "); scanf("%d",&n);
while(*p>1)
{
f=f* *p;
--*p; /* (*p)--; */
}
printf("Factorial=%d",f);
getch();
}
```

The Windows taskbar at the bottom shows various application icons and the system clock indicating 2:52 PM on 07-Oct-23.

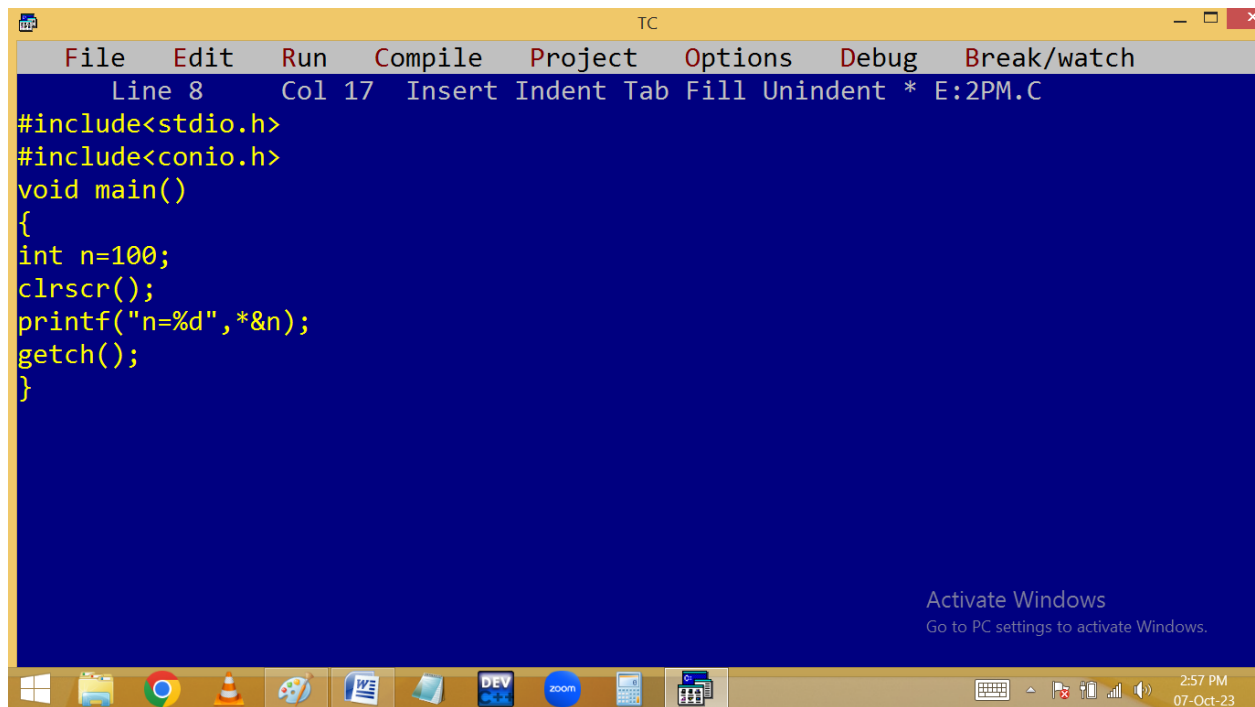


The screenshot shows the Turbo C++ (TC) IDE after the program has been executed. The output window displays the following text:

```
Enter n values 3
Factorial=6
```

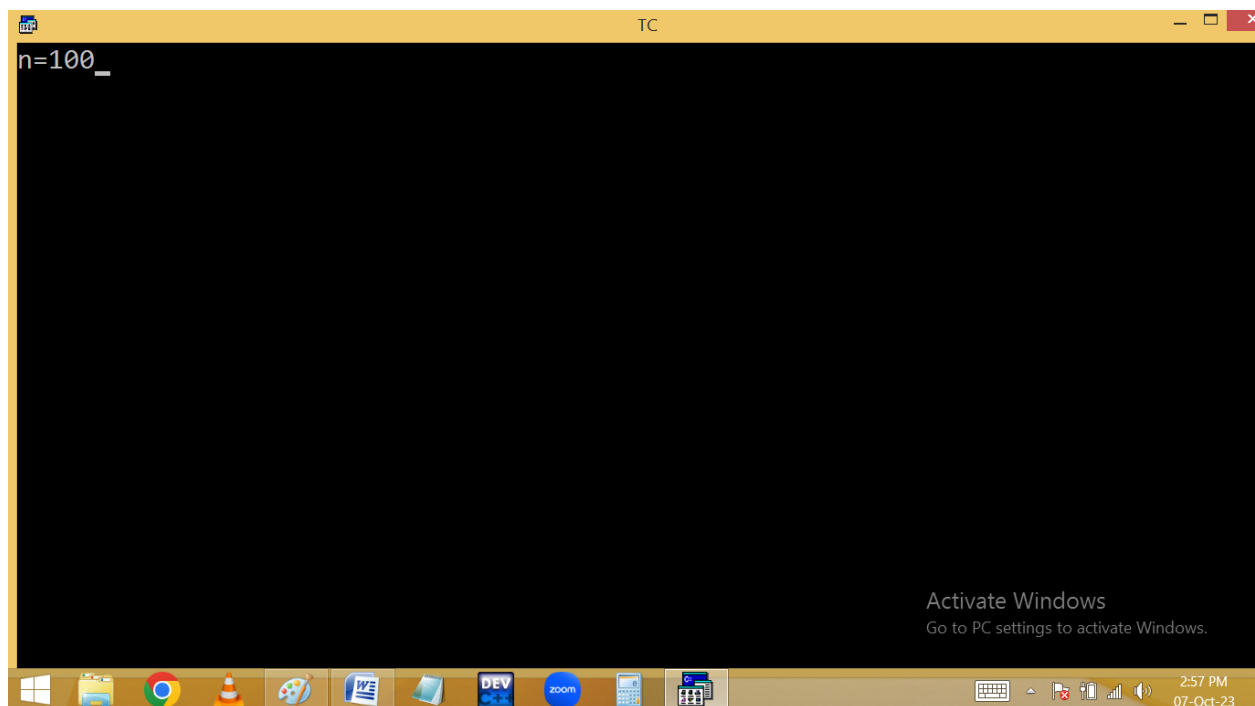
The Windows taskbar at the bottom remains the same, showing the system clock at 2:52 PM on 07-Oct-23.

**Find a normal variable value using pointer mechanism:**



```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 17 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n=100;
clrscr();
printf("n=%d", *&n);
getch();
}

Activate Windows
Go to PC settings to activate Windows.
2:57 PM
07-Oct-23
```

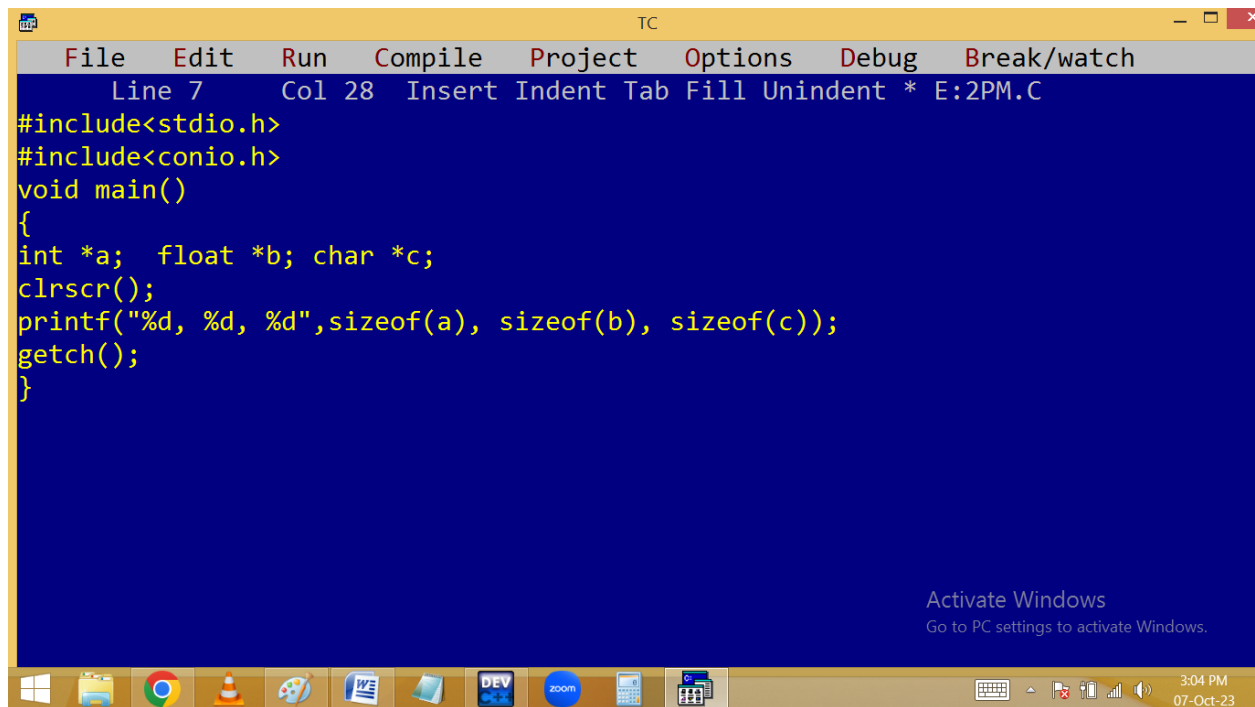


```
TC
n=100_

Activate Windows
Go to PC settings to activate Windows.
2:57 PM
07-Oct-23
```

**Finding pointer size:**

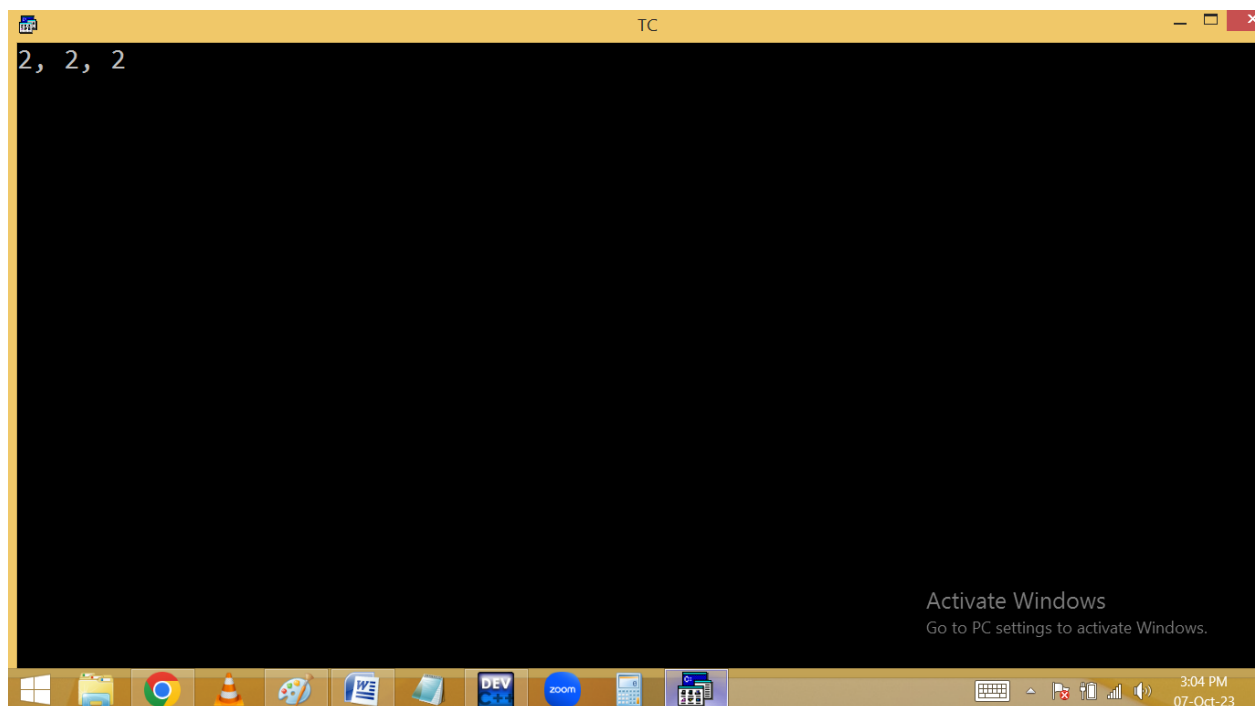
**Pointer always stores the address and address is an unsigned int. due to this any type of pointer it takes 2 / 4 / 8 bytes in 16 / 32 / 64 bit compilers.**



The screenshot shows the Turbo C++ (TC) IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 7, Col 28, Insert, Indent, Tab, Fill, Unindent, \* E:2PM.C). The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
    int *a; float *b; char *c;
    clrscr();
    printf("%d, %d, %d", sizeof(a), sizeof(b), sizeof(c));
    getch();
}
```

The Windows taskbar at the bottom shows the Start button and several application icons, including File Explorer, Google Chrome, VLC media player, Paint, Word, and the Turbo C++ IDE. The system clock in the bottom right corner displays 3:04 PM on 07-Oct-23.

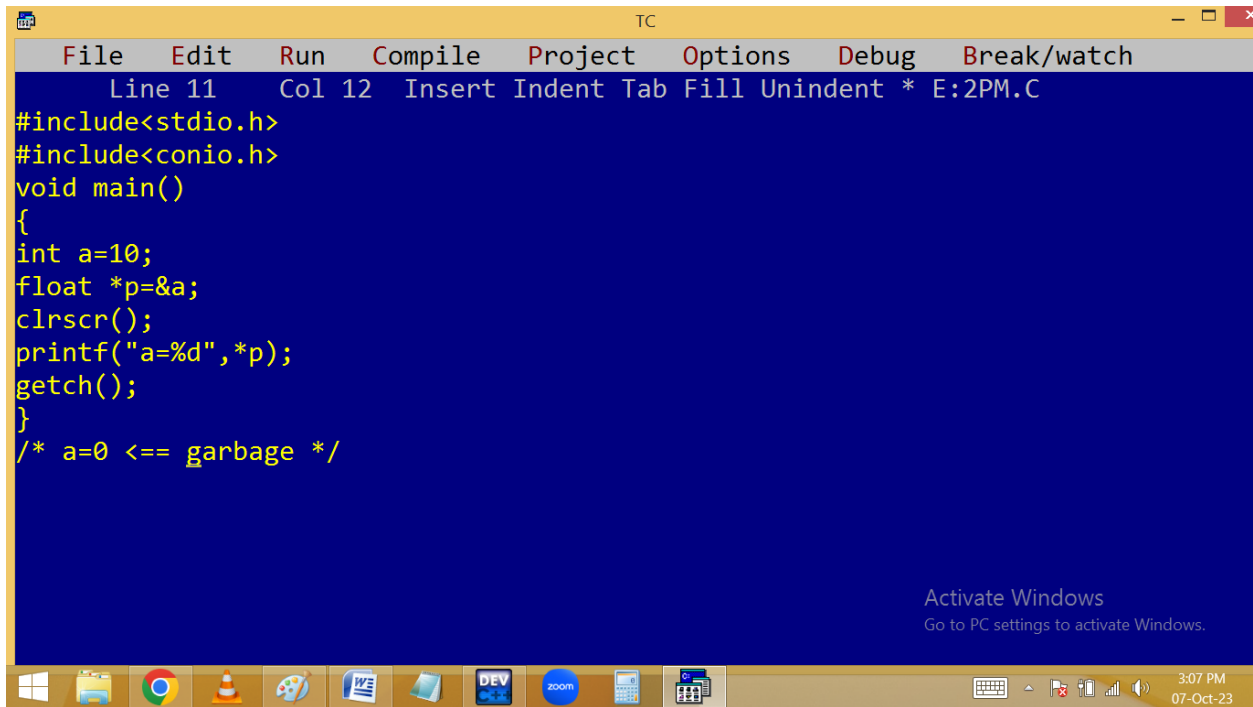


The screenshot shows the Turbo C++ (TC) IDE with the same menu bar and status bar as the first image. The output window at the top left displays the result of the program's execution:

```
2, 2, 2
```

The rest of the IDE interface, including the taskbar and system clock, remains the same as in the first image.

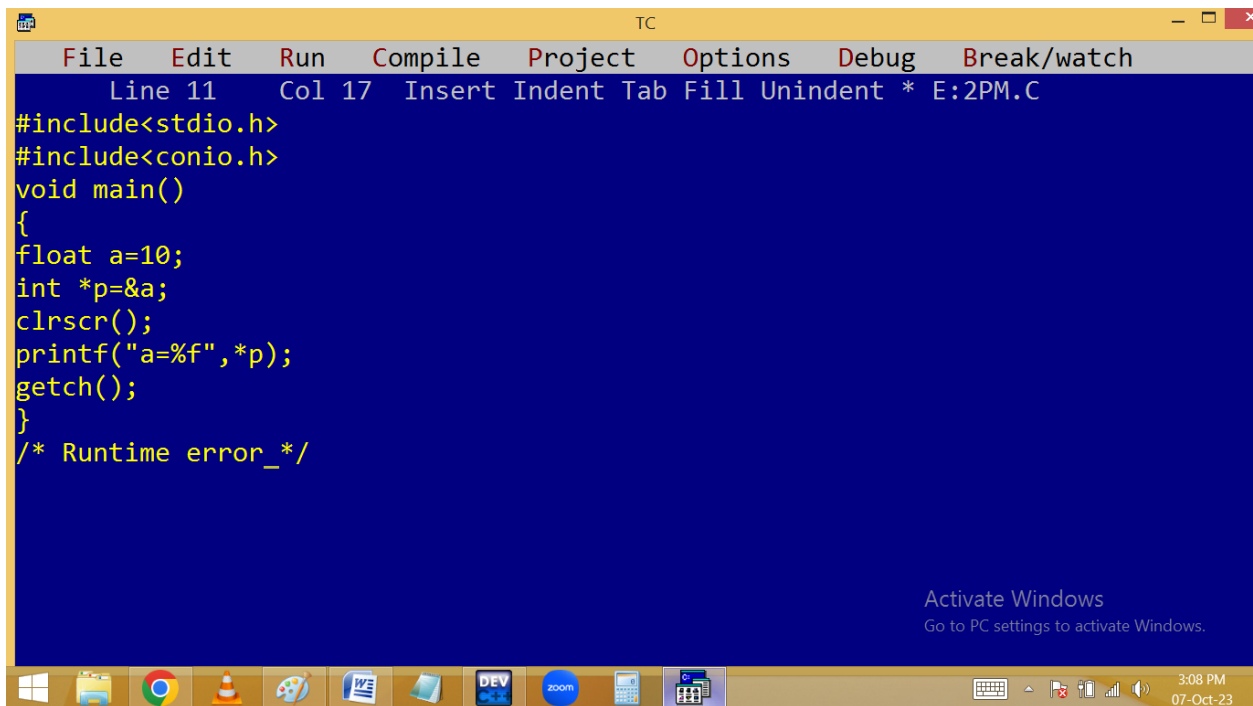
**Pointer compatibility:**



The screenshot shows the Turbo C++ IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 11, Col 12, Insert, Indent, Tab, Fill, Unindent, \* E:2PM.C). The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10;
float *p=&a;
clrscr();
printf("a=%d",*p);
getch();
}
/* a=0 <== garbage */
```

An "Activate Windows" watermark is visible in the bottom right corner of the IDE window. The Windows taskbar at the bottom shows icons for File Explorer, Chrome, VLC, Paint, Word, and several other applications, along with the system clock showing 3:07 PM on 07-Oct-23.



The screenshot shows the Turbo C++ IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 11, Col 17, Insert, Indent, Tab, Fill, Unindent, \* E:2PM.C). The code in the editor is as follows:

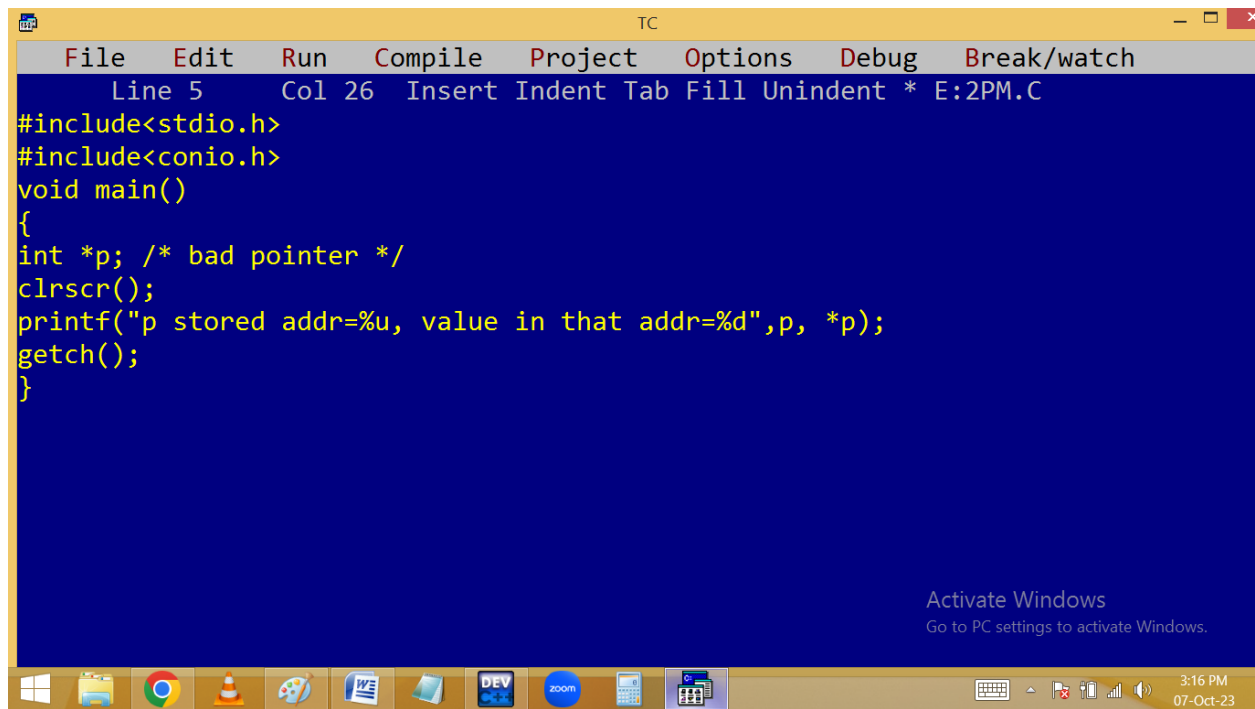
```
#include<stdio.h>
#include<conio.h>
void main()
{
float a=10;
int *p=&a;
clrscr();
printf("a=%f",*p);
getch();
}
/* Runtime error_*/
```

An "Activate Windows" watermark is visible in the bottom right corner of the IDE window. The Windows taskbar at the bottom shows icons for File Explorer, Chrome, VLC, Paint, Word, and several other applications, along with the system clock showing 3:08 PM on 07-Oct-23.

### **Bad / wild pointer:**

**A pointer is declared but not initialized. In this situation the pointer is taking some unknown address and value. This kind of pointer is called bad / wild pointer.**

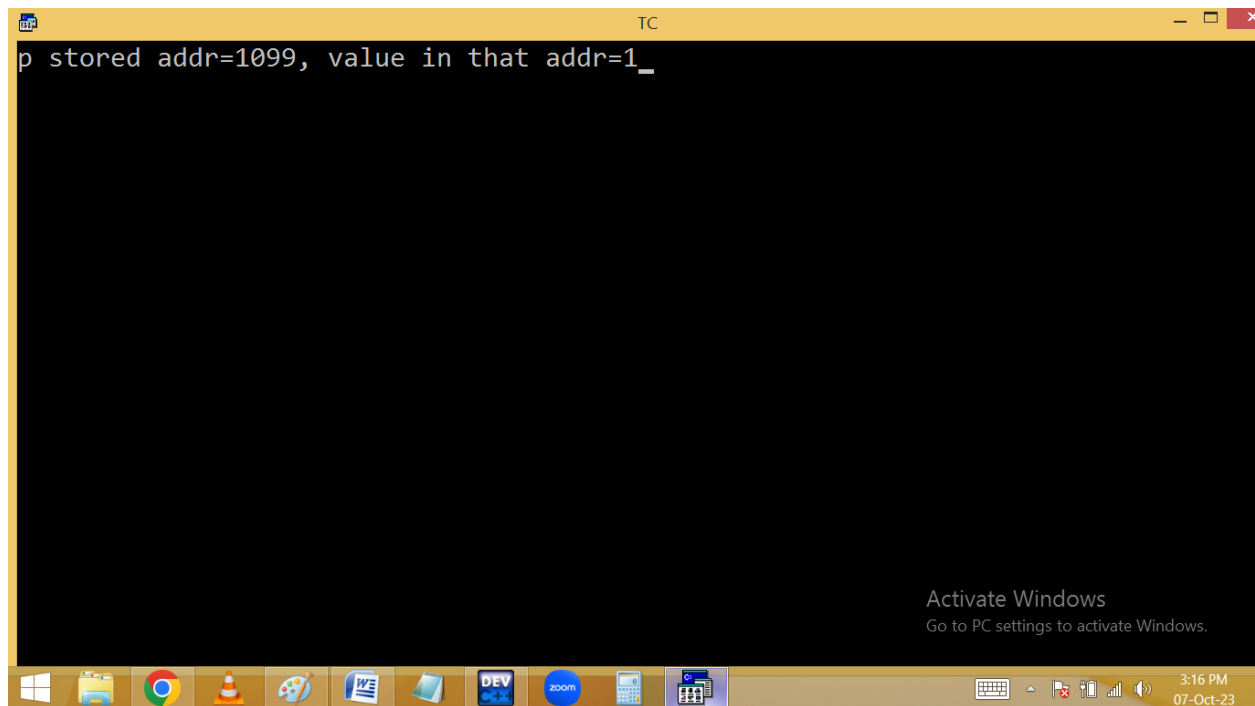




```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 26 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int *p; /* bad pointer */
clrscr();
printf("p stored addr=%u, value in that addr=%d",p, *p);
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

3:16 PM  
07-Oct-23



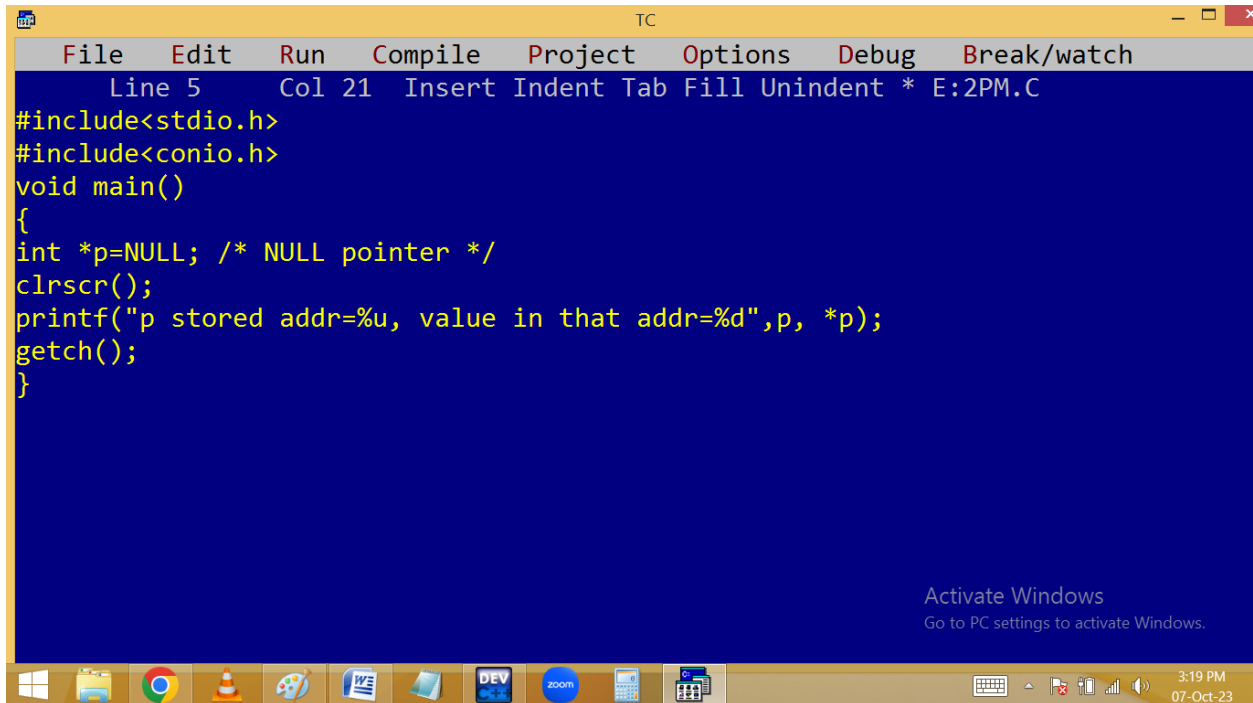
```
TC
p stored addr=1099, value in that addr=1_

```

Activate Windows  
Go to PC settings to activate Windows.

3:16 PM  
07-Oct-23

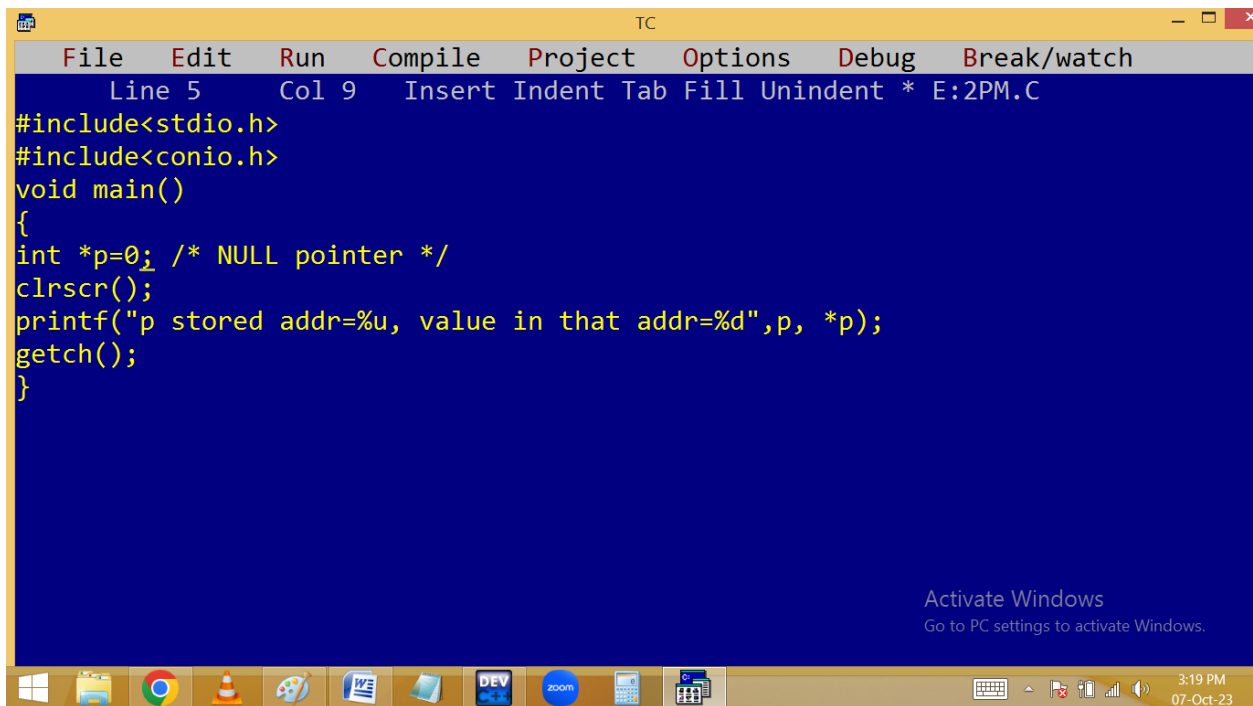
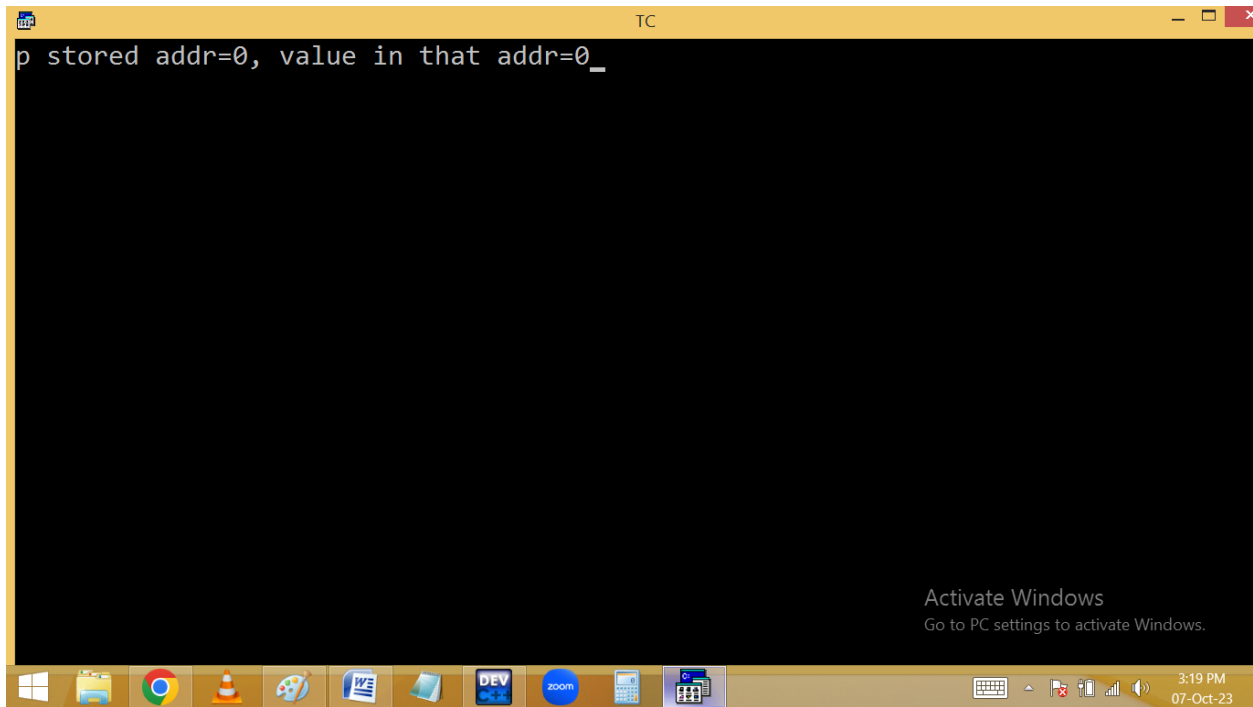
**NULL pointer:** When a pointer is initialized with NULL / 0 then it is called NULL pointer. To avoid bad and dangling pointers we need NULL pointer.

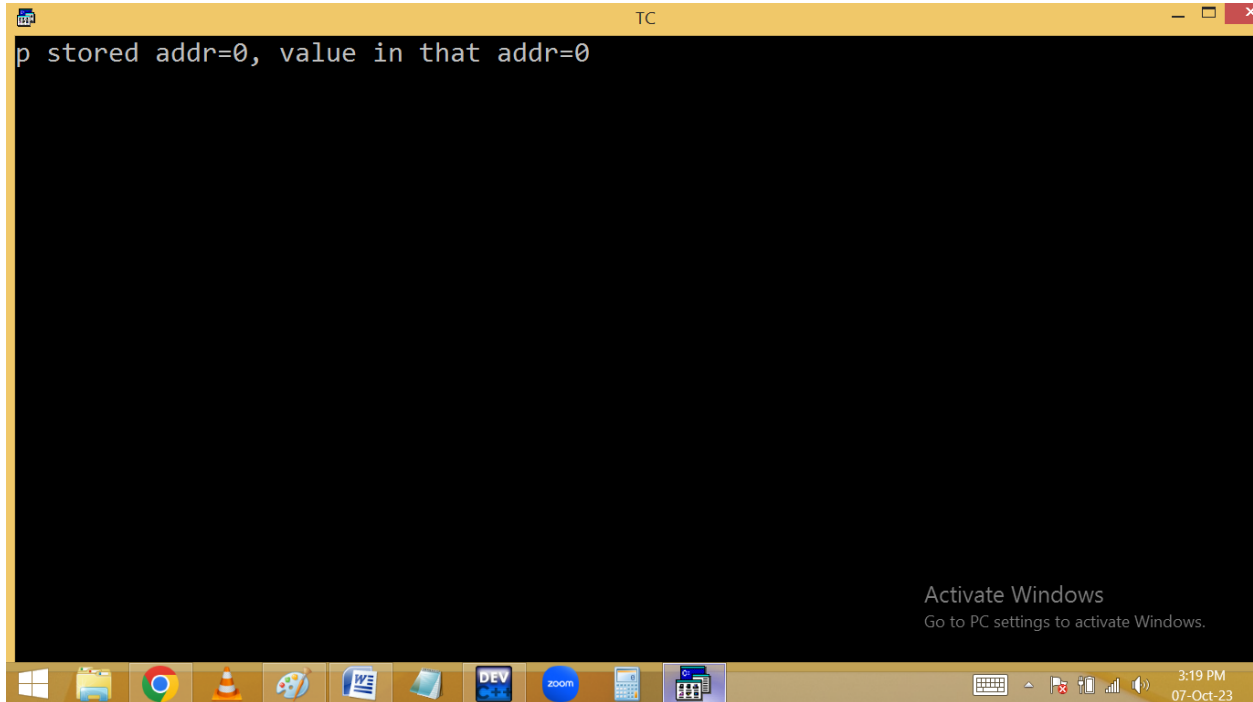


```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 21 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int *p=NULL; /* NULL pointer */
clrscr();
printf("p stored addr=%u, value in that addr=%d",p, *p);
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

3:19 PM  
07-Oct-23



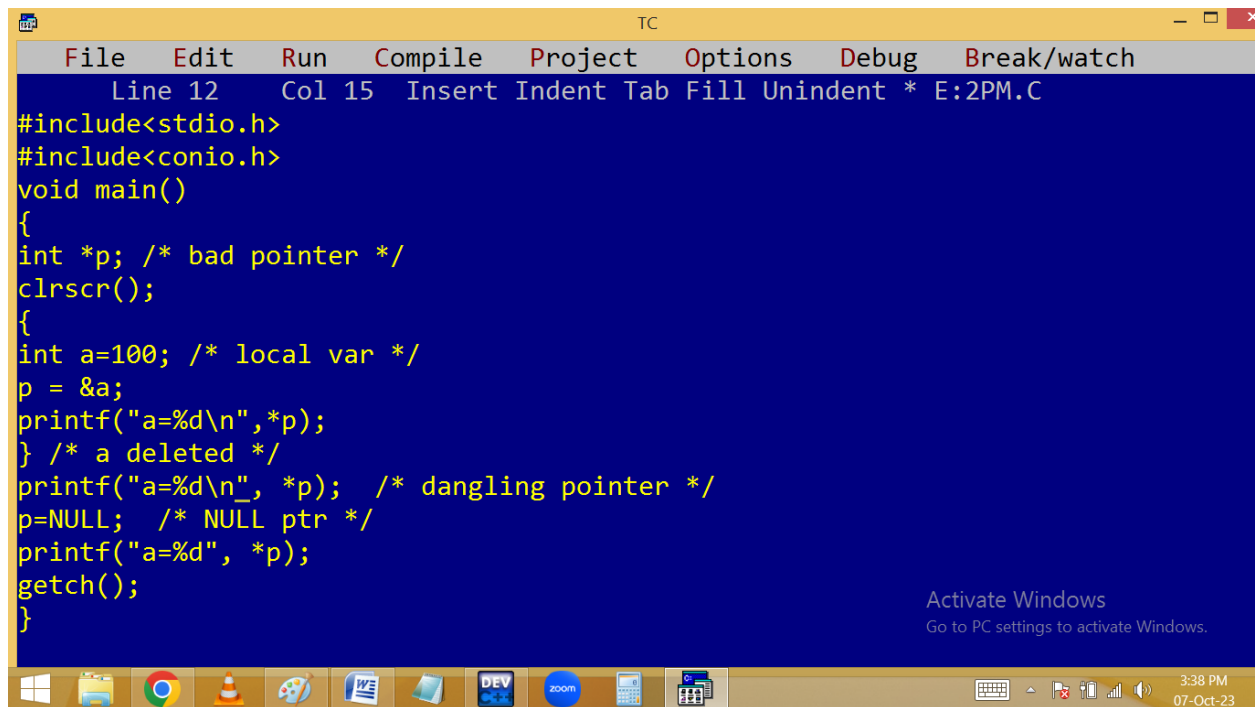


The image shows a screenshot of a Turbo C++ (TC) compiler window. The window has a yellow title bar with the text "TC" in the center. The main area is black and contains the text "p stored addr=0, value in that addr=0" in white. At the bottom of the window is a taskbar with various icons including Windows, File Explorer, Google Chrome, VLC media player, Paint, Word, a folder, DEV, Zoom, and a calculator. The system tray on the right shows the time "3:19 PM" and the date "07-Oct-23". A watermark "Activate Windows" is visible in the bottom right corner of the window.

```
p stored addr=0, value in that addr=0
```

## **Dangling pointer:**

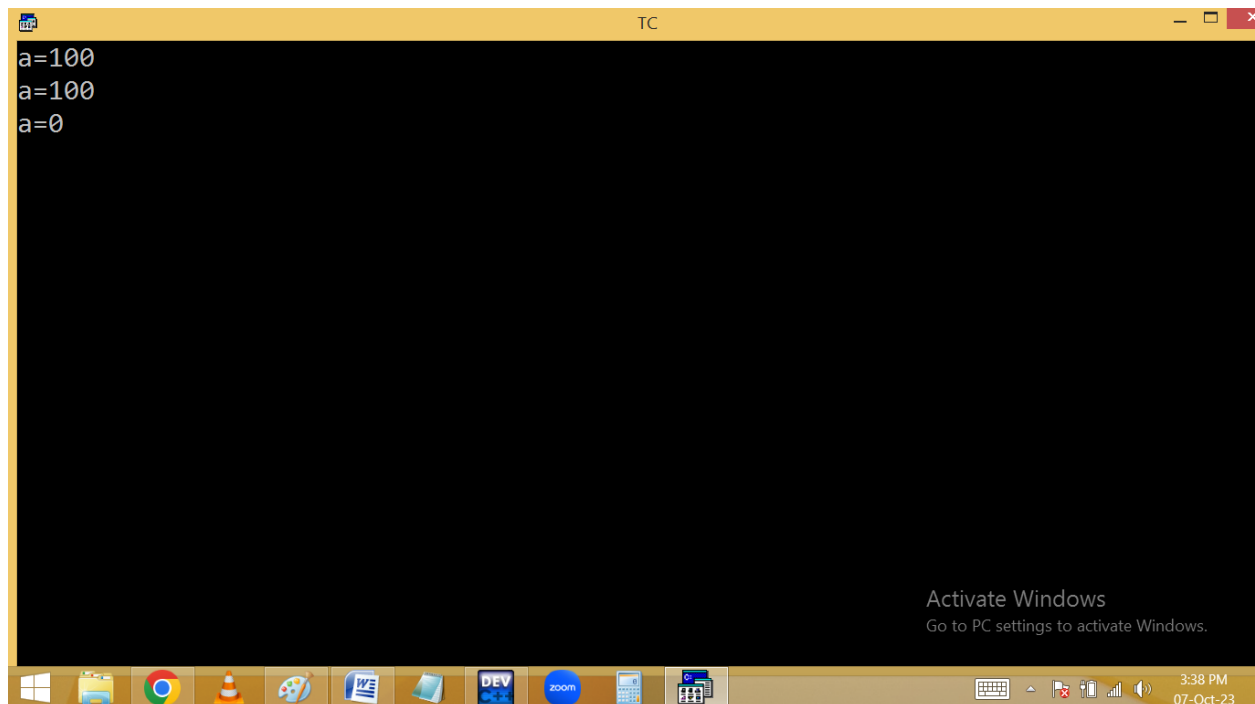
A pointer is declared and some variable address also assigned. After some time that variable deleted from memory. After deleting the variable also the pointer is still holding the address of that deleted variable. This kind of pointer is called dangling pointer and to avoid this initialize with NULL.



The screenshot shows the Turbo C++ (TC) IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 12, Col 15, Insert, Indent, Tab, Fill, Unindent, \* E:2PM.C). The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *p; /* bad pointer */
clrscr();
{
int a=100; /* local var */
p = &a;
printf("a=%d\n",*p);
} /* a deleted */
printf("a=%d\n_", *p); /* dangling pointer */
p=NULL; /* NULL ptr */
printf("a=%d", *p);
getch();
}
```

An "Activate Windows" watermark is visible in the bottom right corner of the IDE window. The Windows taskbar at the bottom shows various application icons and the system clock indicating 3:38 PM on 07-Oct-23.



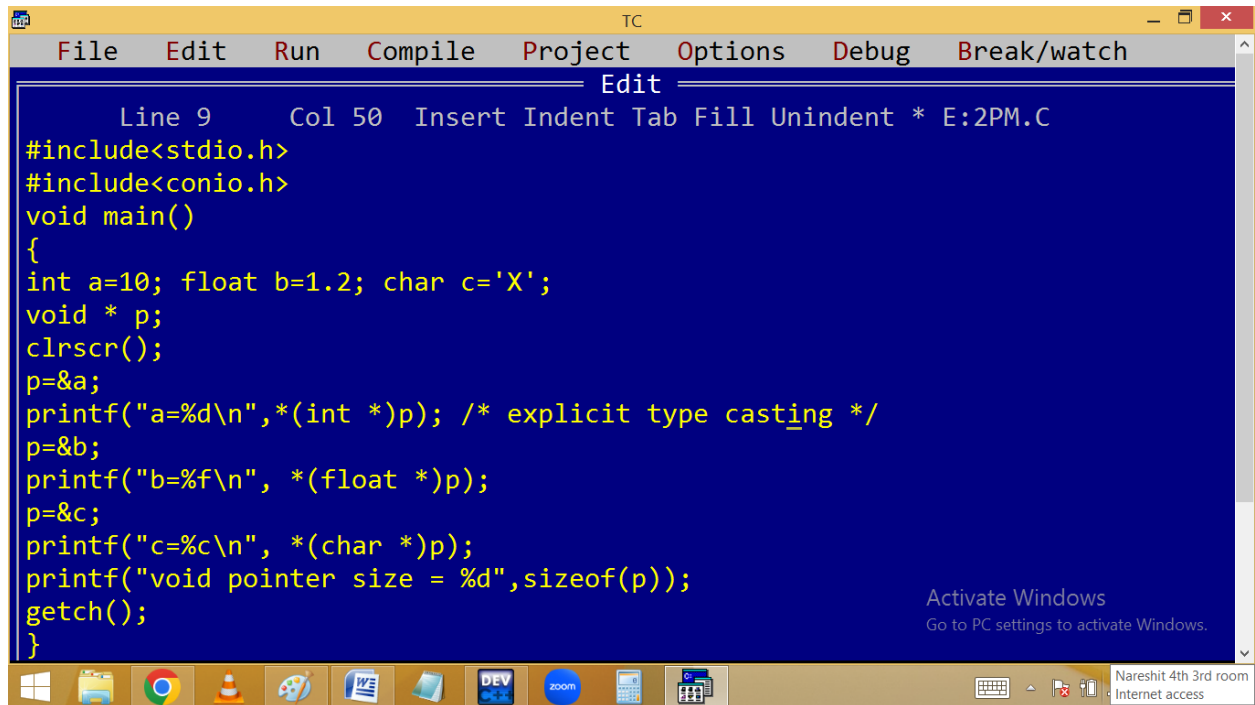
The screenshot shows the Turbo C++ (TC) IDE with the same menu bar and status bar. The output window displays the following text:

```
a=100
a=100
a=0
```

An "Activate Windows" watermark is visible in the bottom right corner of the IDE window. The Windows taskbar at the bottom shows various application icons and the system clock indicating 3:38 PM on 07-Oct-23.

**void / generic pointer:** pointer stores same type of variable address. When several variables with different

**data types, use void pointer to store the address. Void pointer can store any type of address. But before using void pointer, explicit type casting should be provided. Void pointer takes 2 bytes and it is very much used in dynamic memory allocation.**



TC

File Edit Run Compile Project Options Debug Break/watch

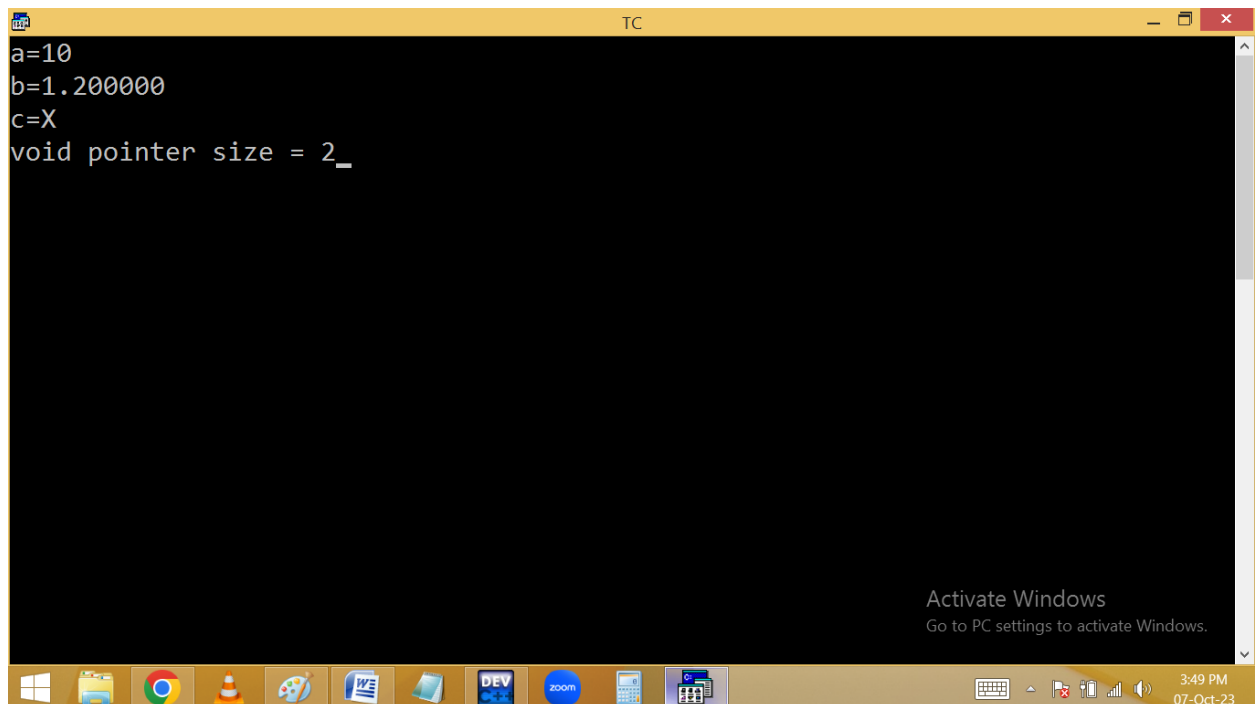
Edit

Line 9 Col 50 Insert Indent Tab Fill Unindent \* E:2PM.C

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10; float b=1.2; char c='X';
void * p;
clrscr();
p=&a;
printf("a=%d\n",*(int *)p); /* explicit type casting */
p=&b;
printf("b=%f\n", *(float *)p);
p=&c;
printf("c=%c\n", *(char *)p);
printf("void pointer size = %d",sizeof(p));
getch();
}
```

Activate Windows  
Go to PC settings to activate Windows.

Nareshit 4th 3rd room  
Internet access



TC

```
a=10
b=1.200000
c=X
void pointer size = 2_
```

Activate Windows  
Go to PC settings to activate Windows.

3:49 PM  
07-Oct-23