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/Reference**].
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.

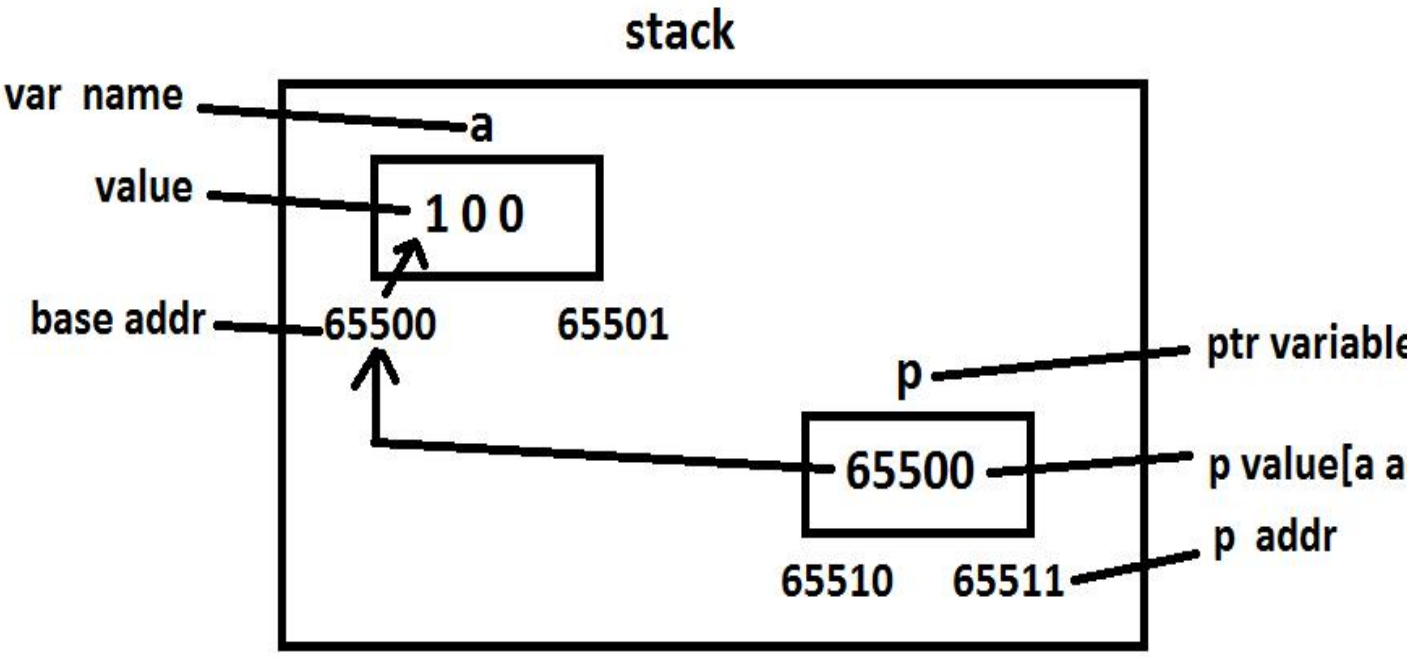
Eg:

```
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**.

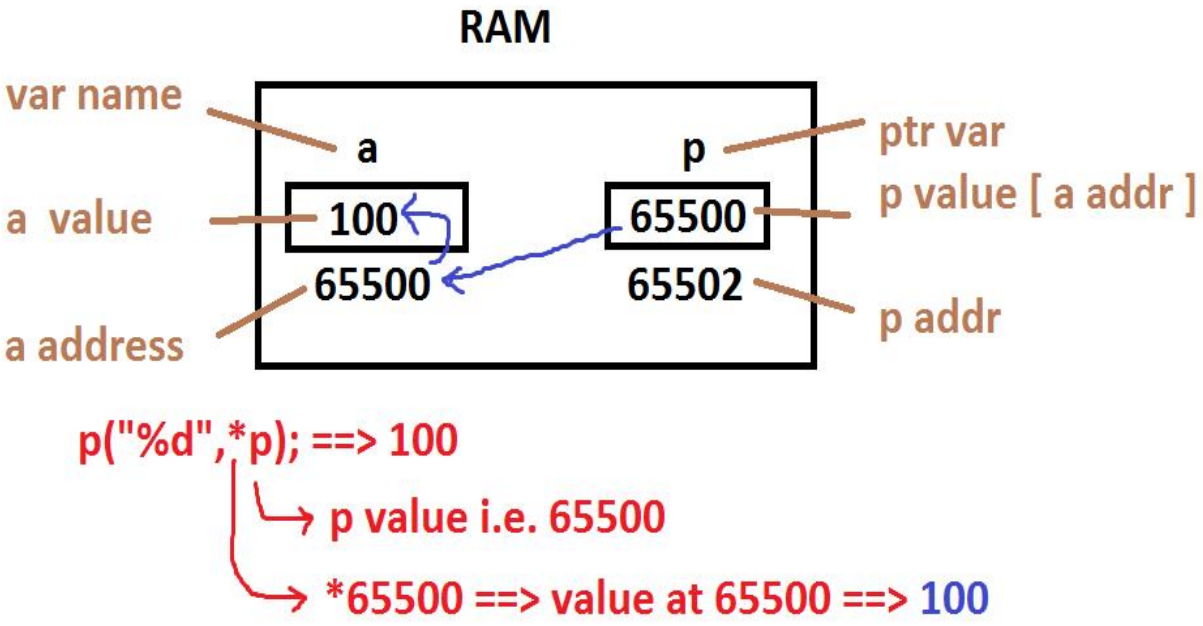
Or

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

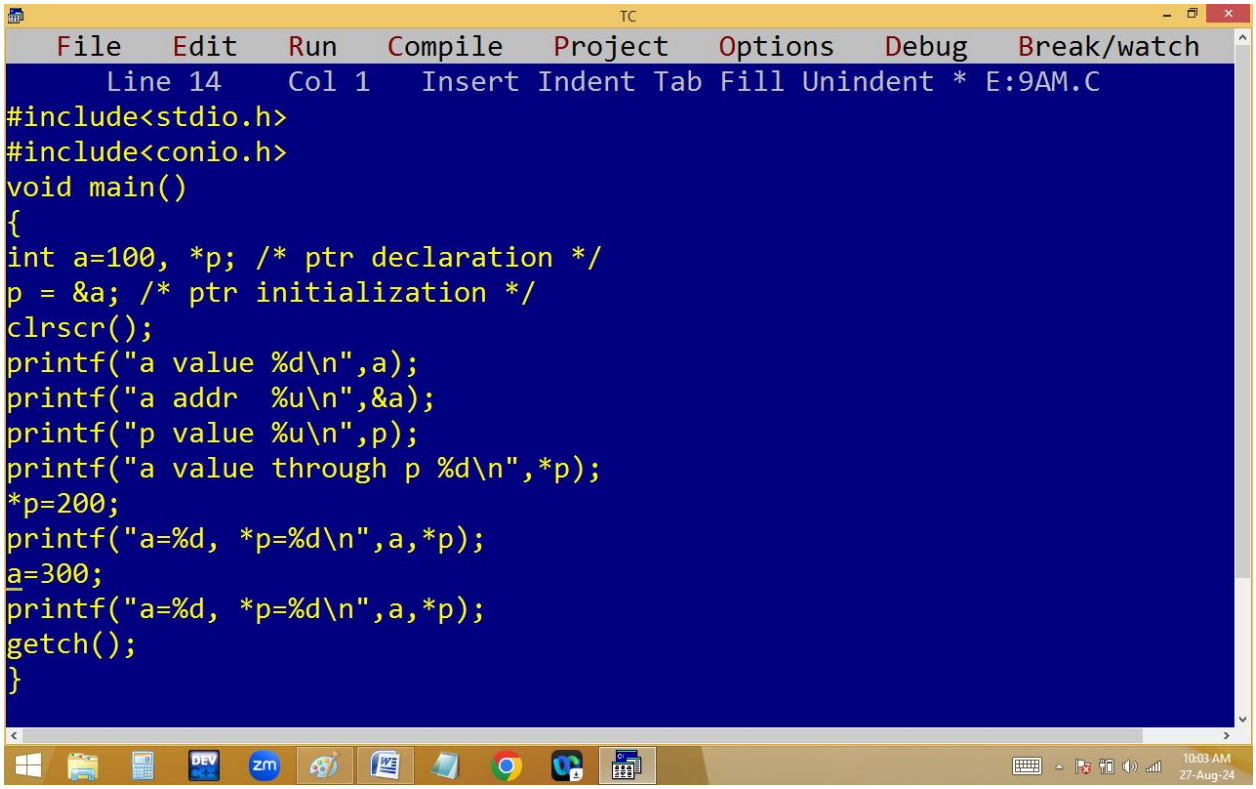
Due to this example any changes conducted in ***p** effects the value of **'a'**. Hence **p** is called **pointer** to **a**.

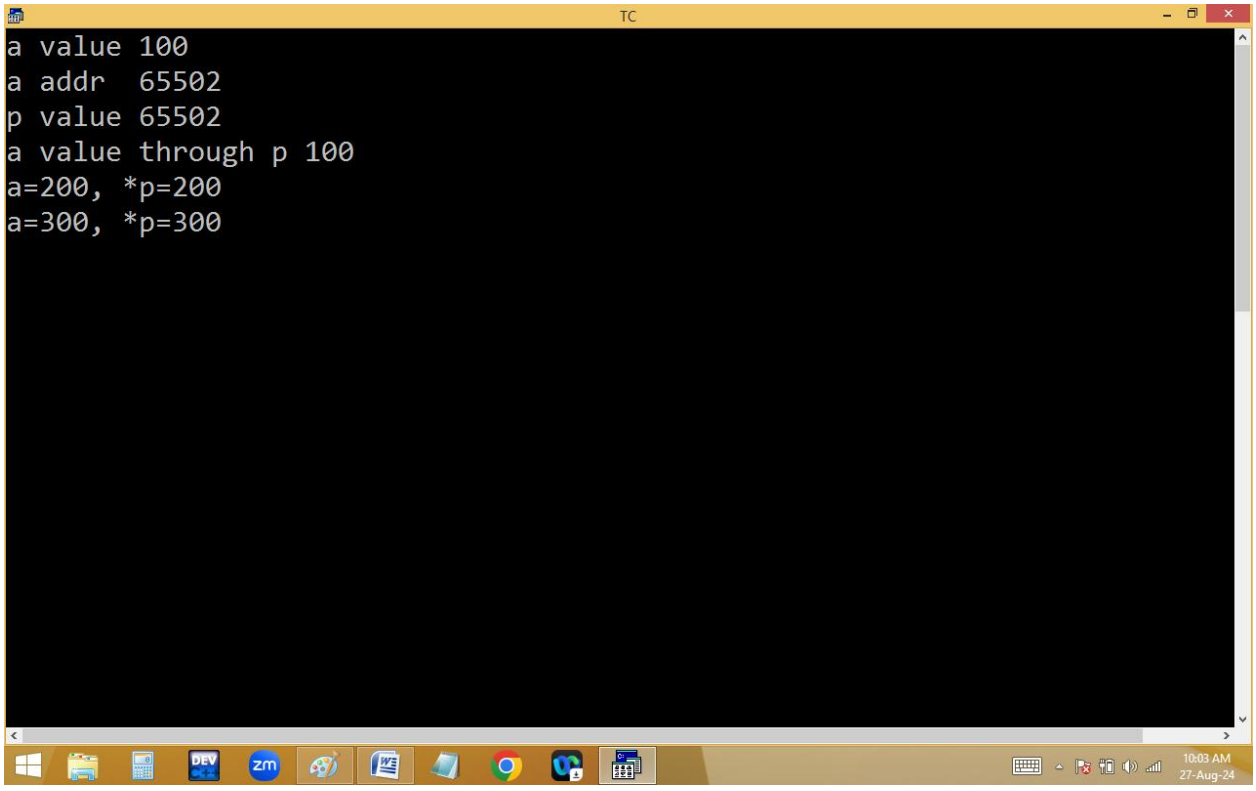
Eg: *p=200;

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

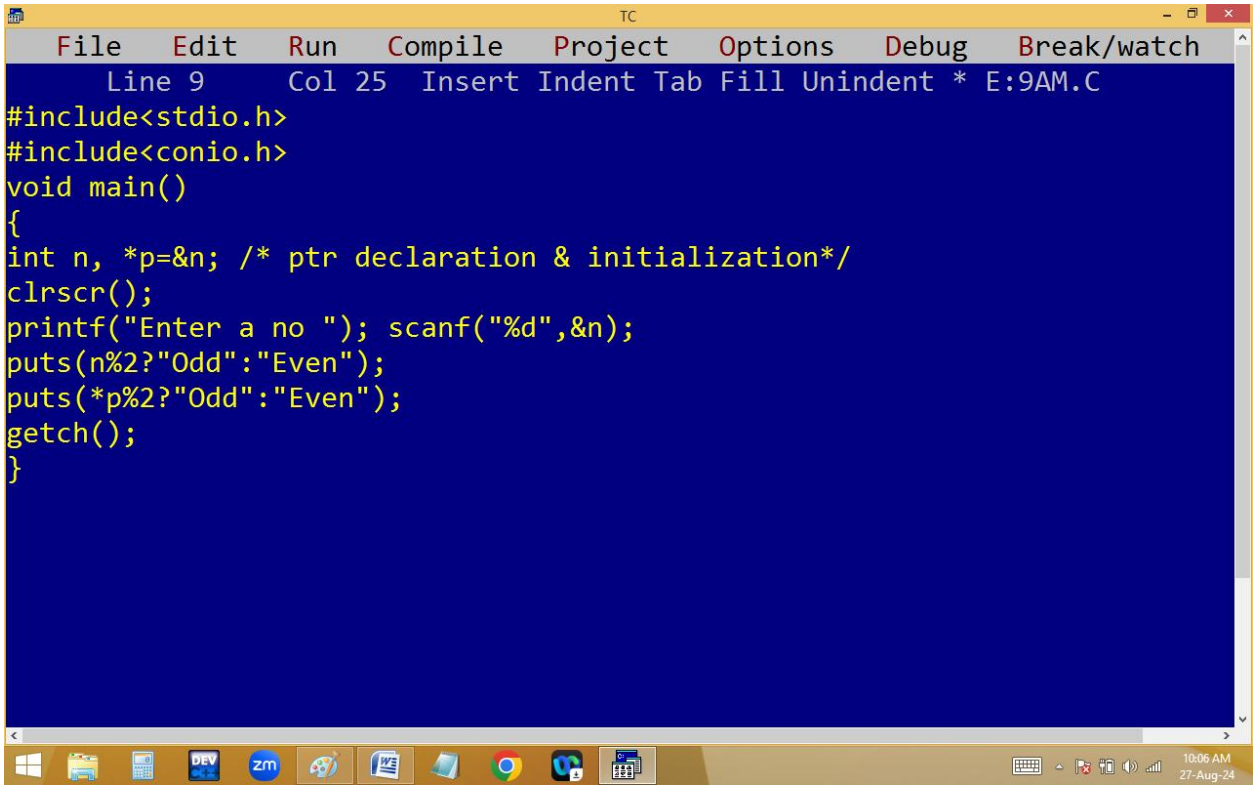


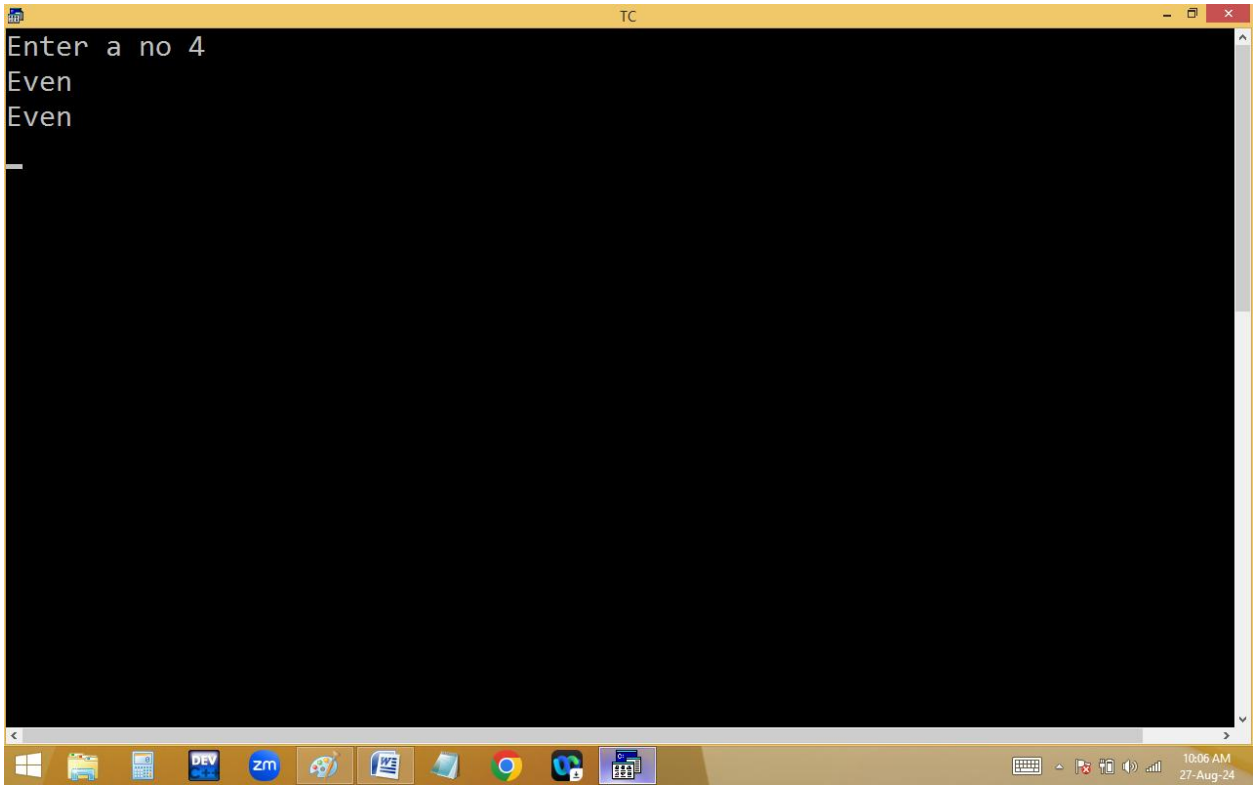
Finding a variable value and address using pointer:

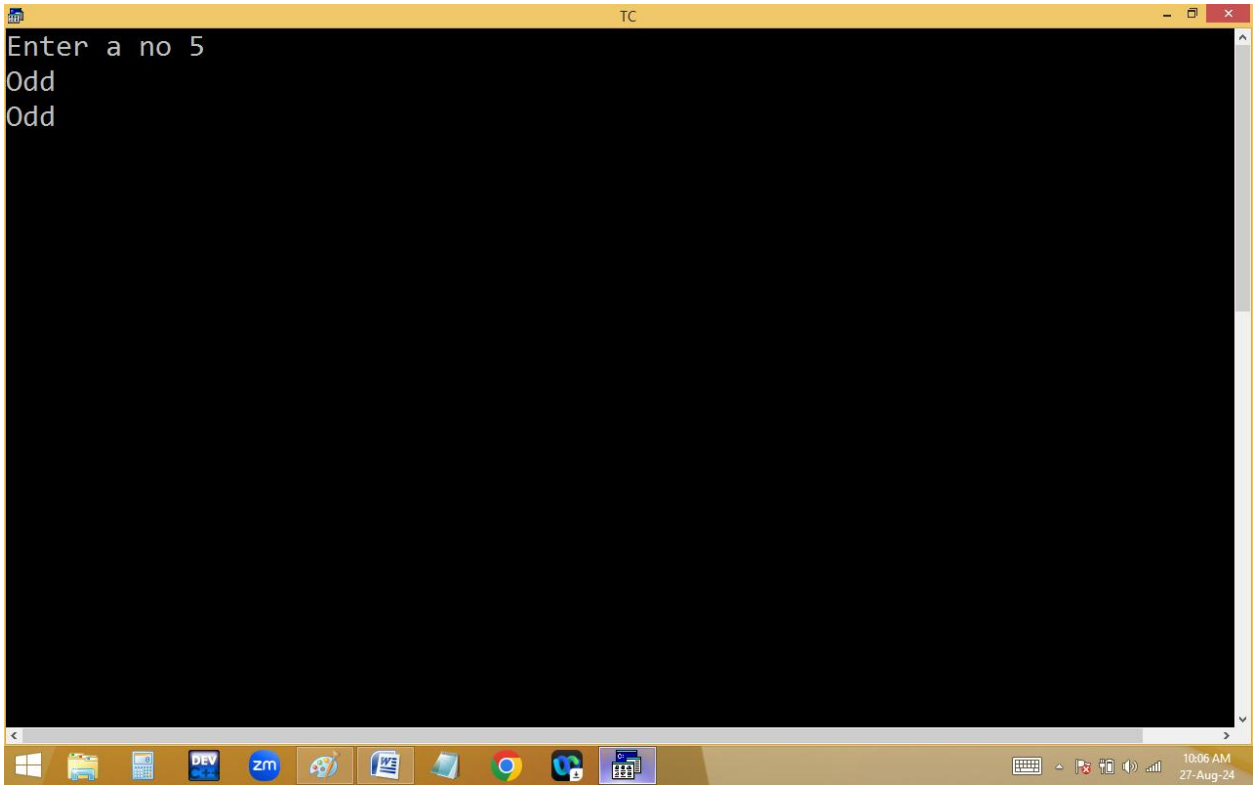




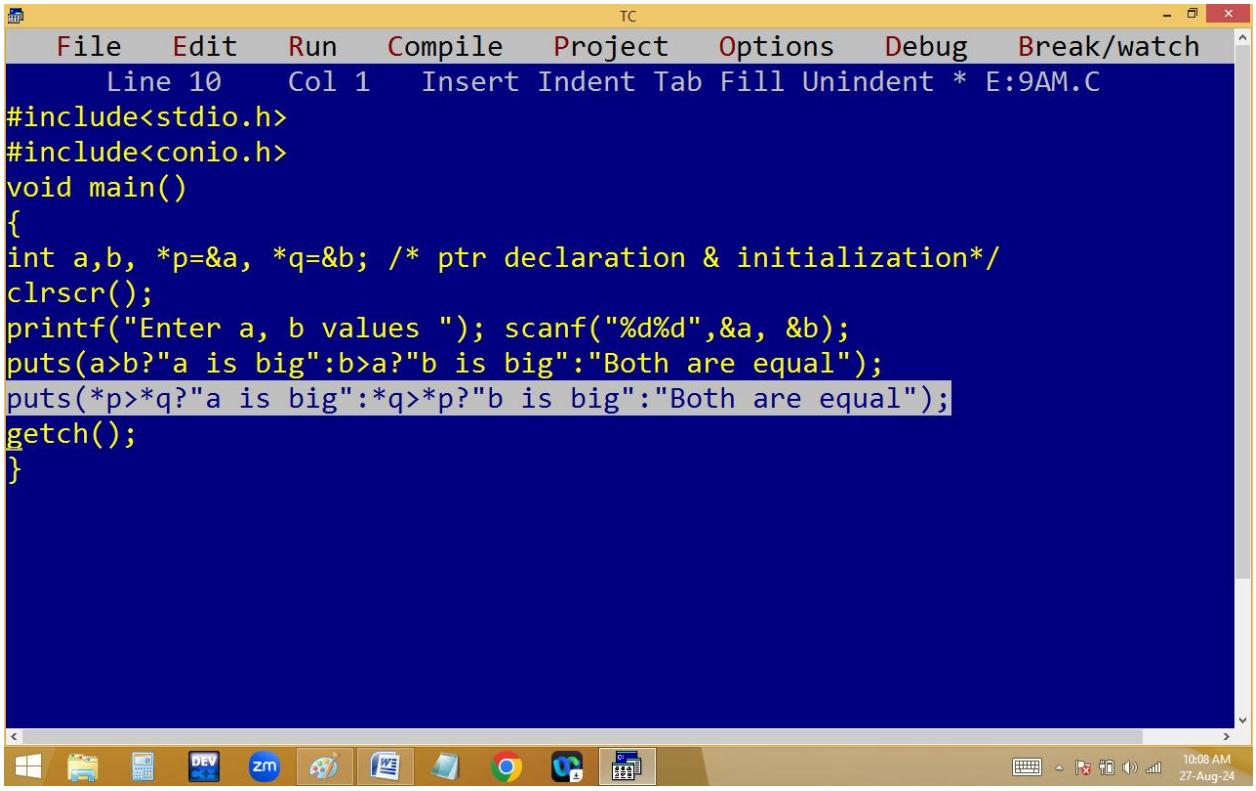
Finding even/odd using pointer:

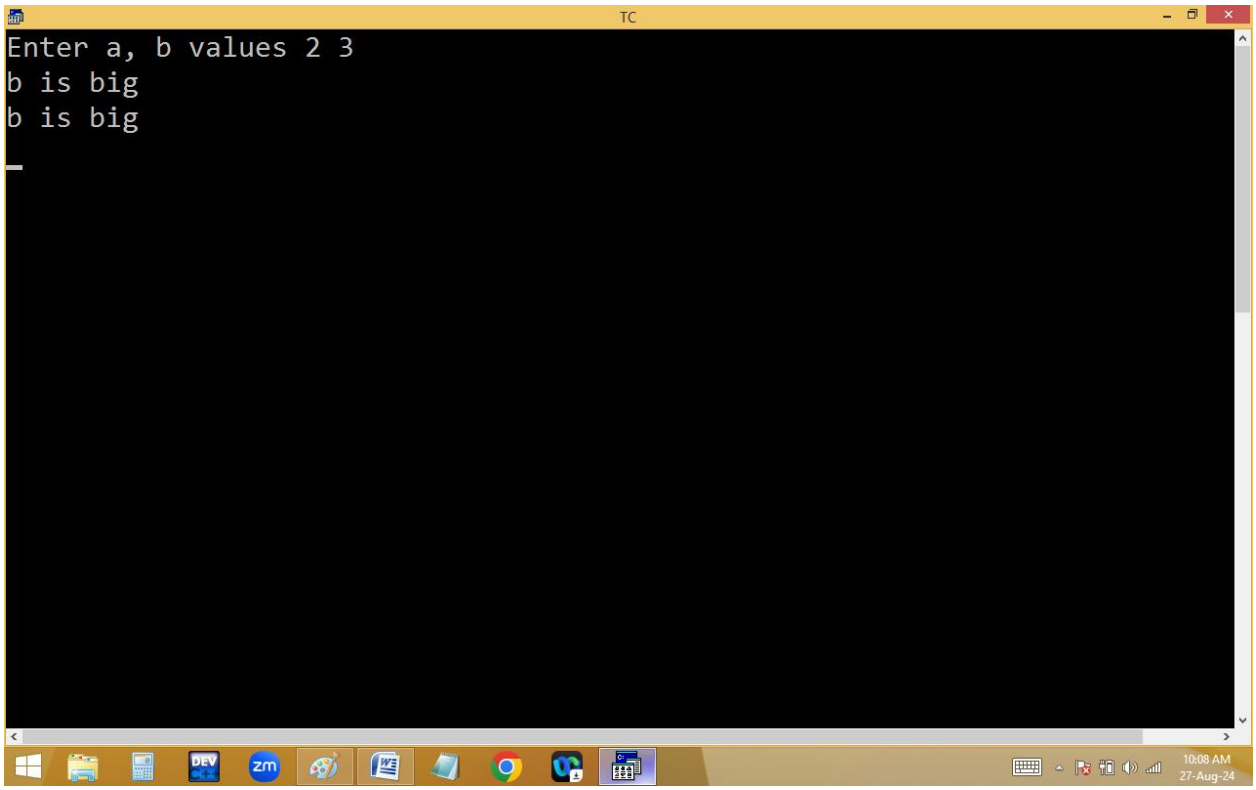


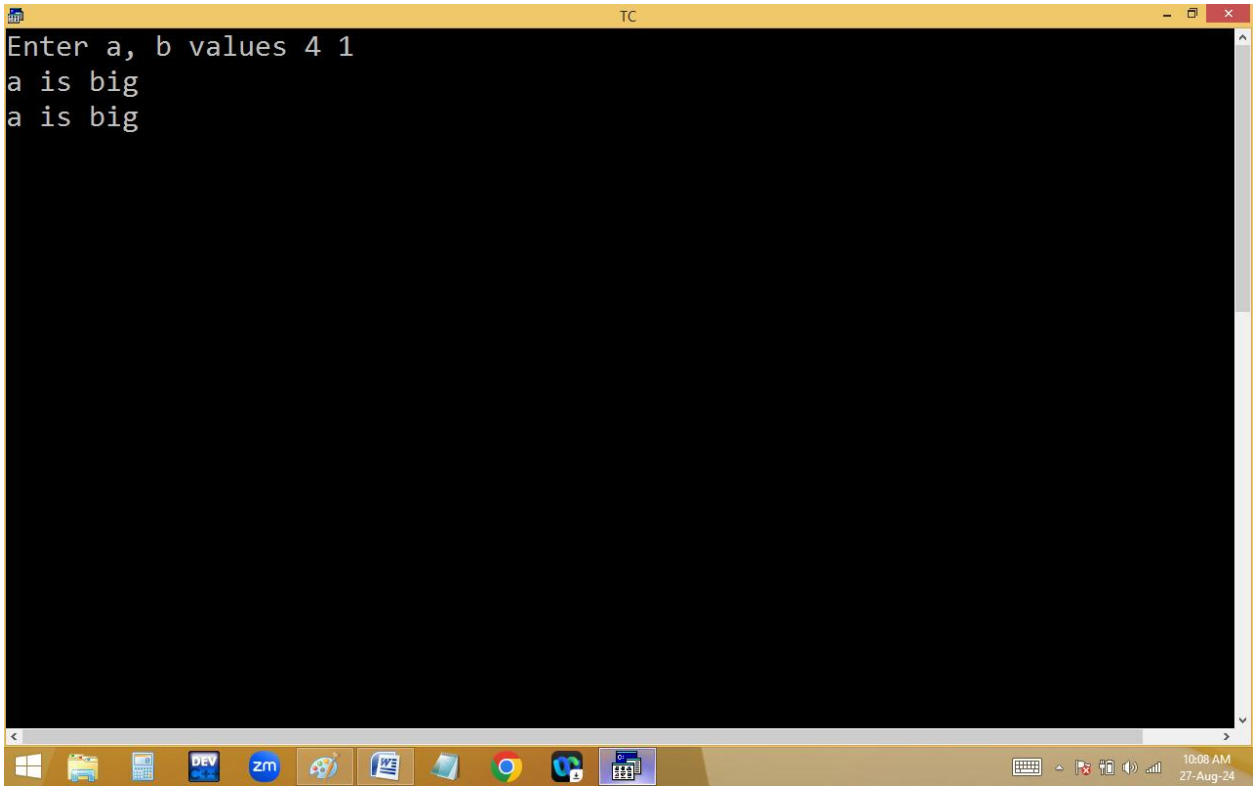


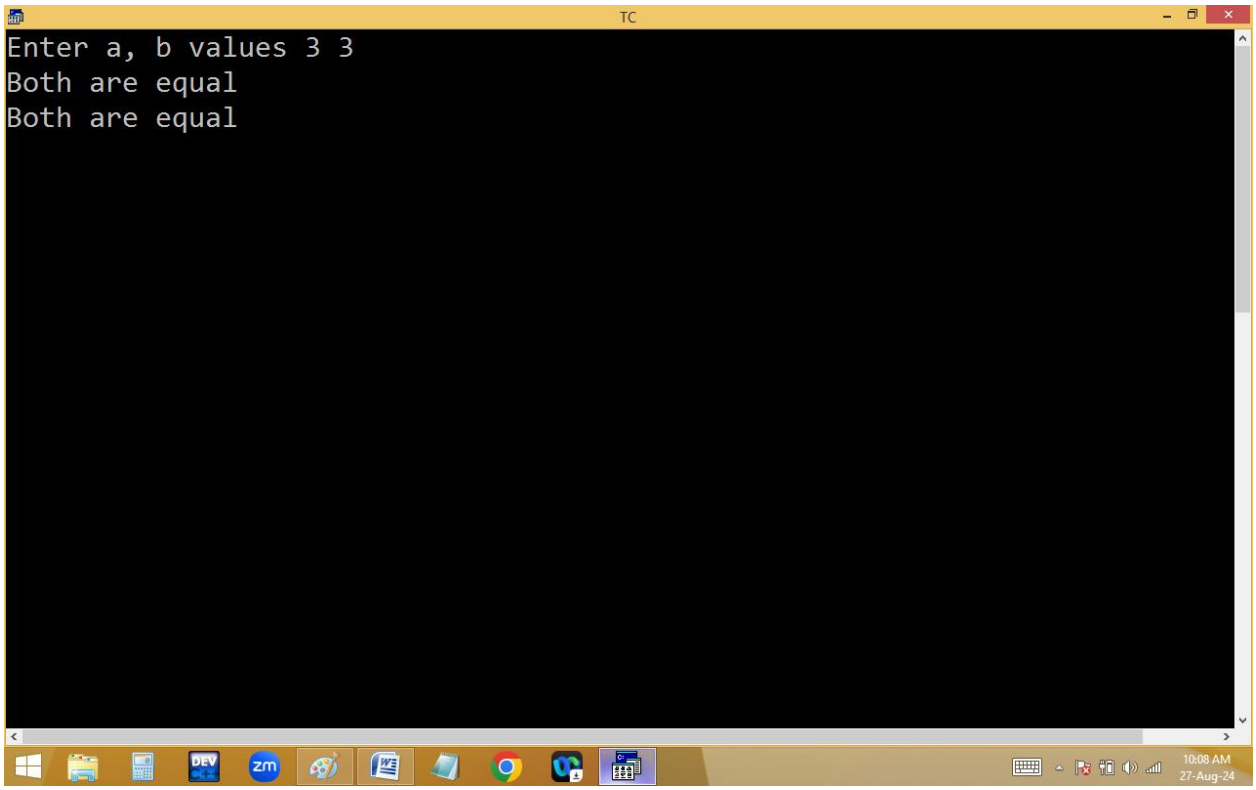


Finding max in 2 no’s using pointers:

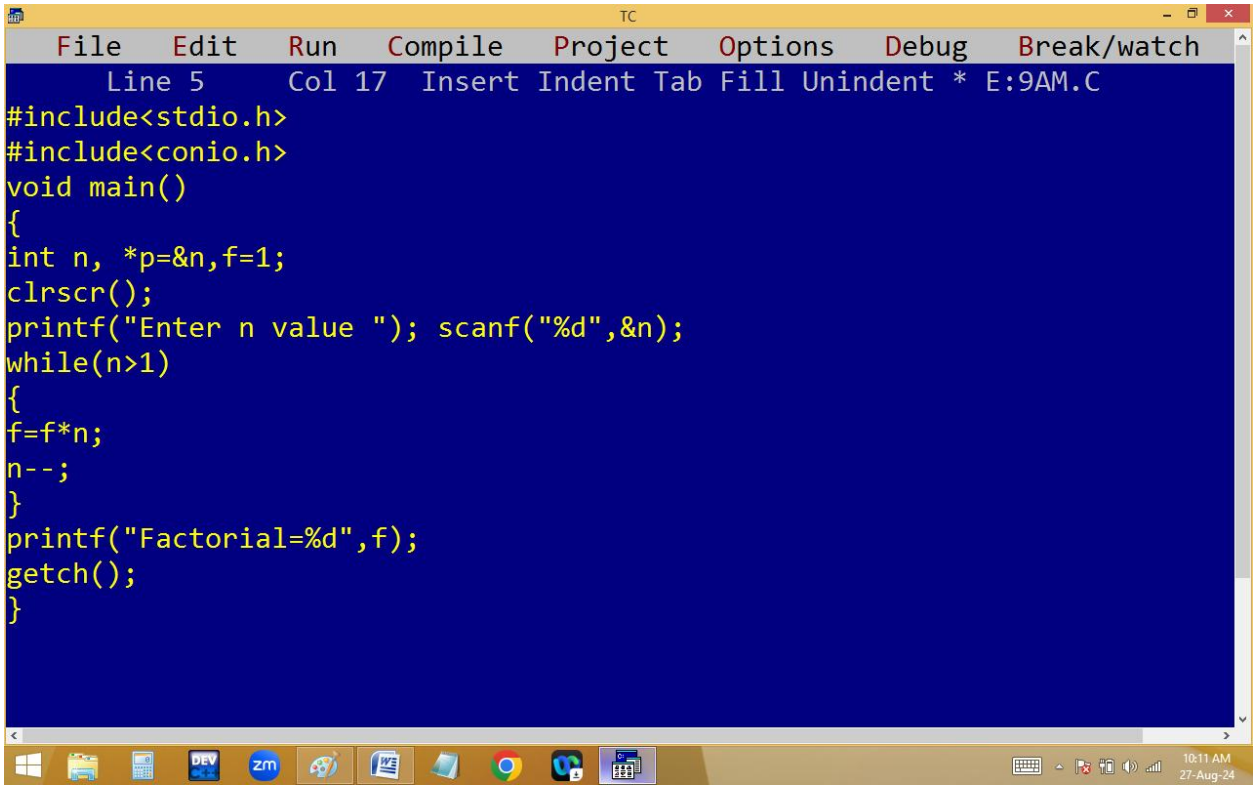

TC
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b, *p=&a, *q=&b; /* ptr declaration & initialization*/
clrscr();
printf("Enter a, b values "); scanf("%d%d",&a, &b);
puts(a>b?"a is big":b>a?"b is big":"Both are equal");
puts(*p>*q?"a is big":*q>*p?"b is big":"Both are equal");
getch();
}

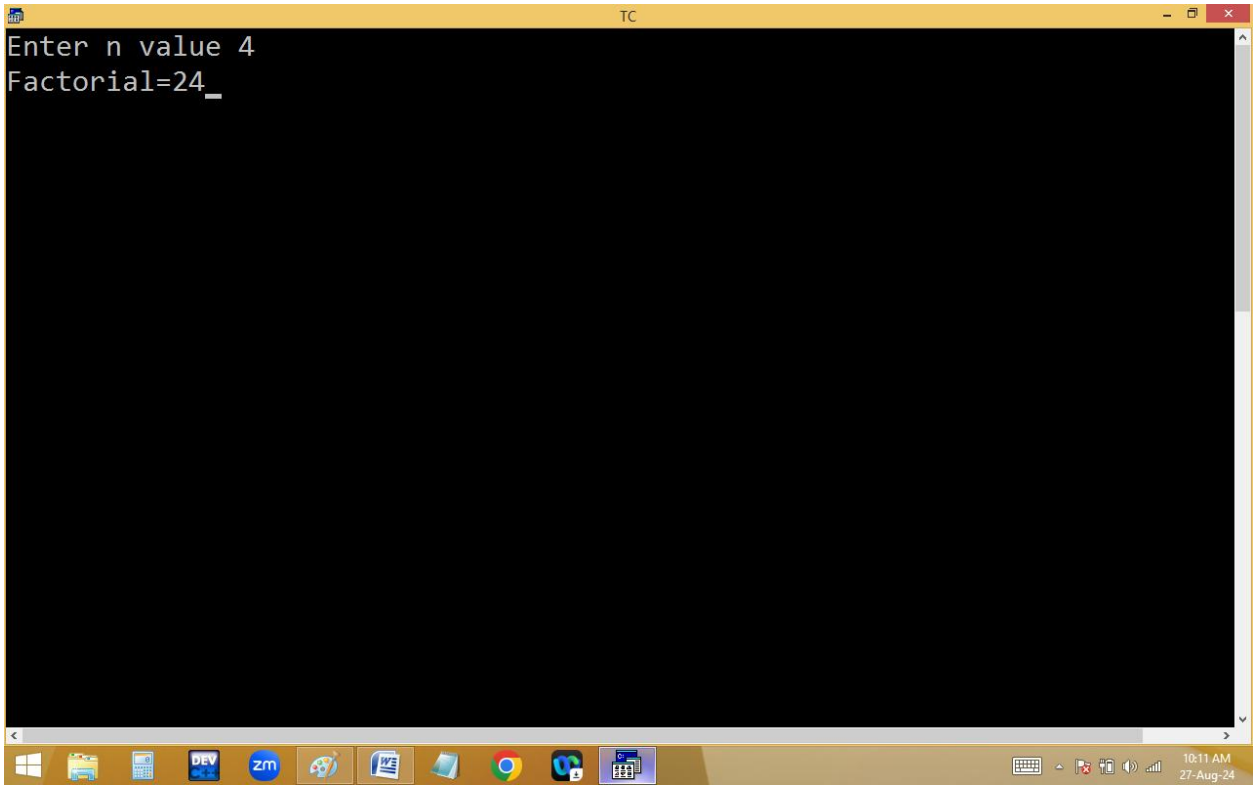

TC
Enter a, b values 2 3
b is big
b is big

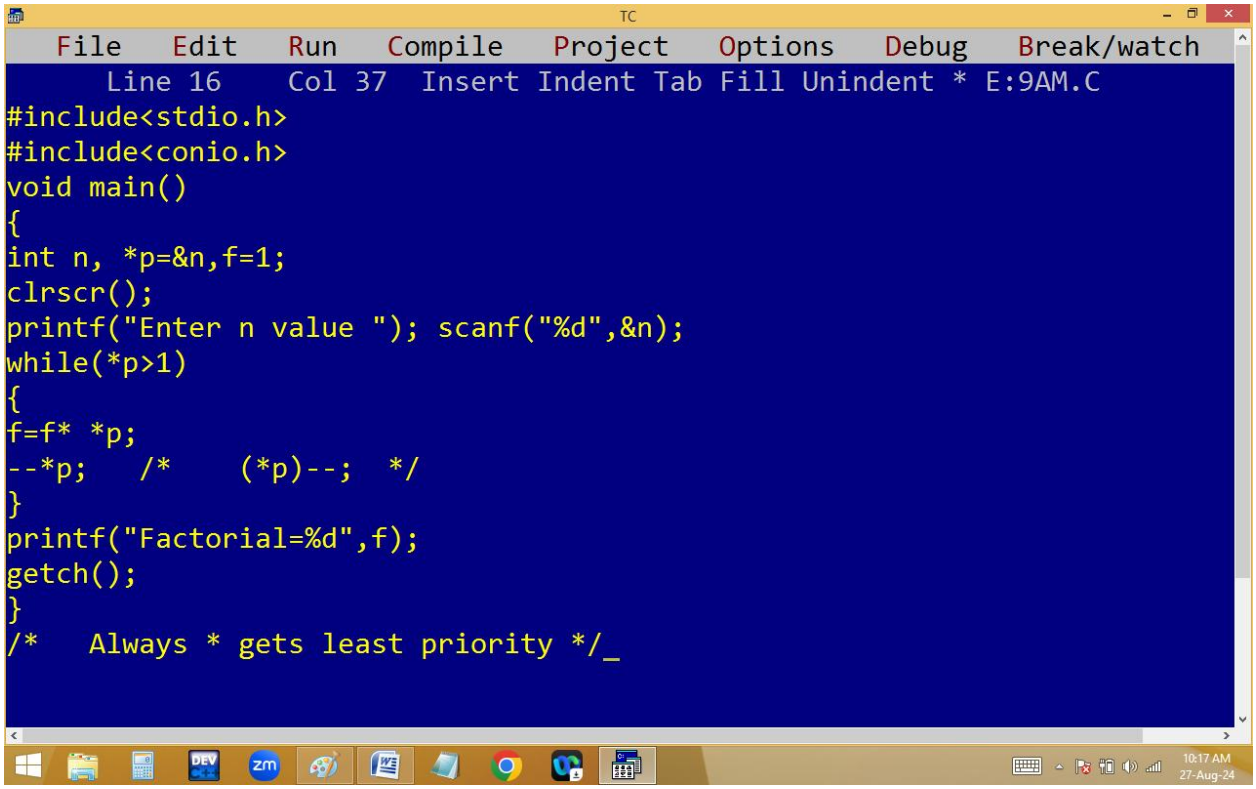

TC
Enter a, b values 4 1
a is big
a is big
10:08 AM
27-Aug-24

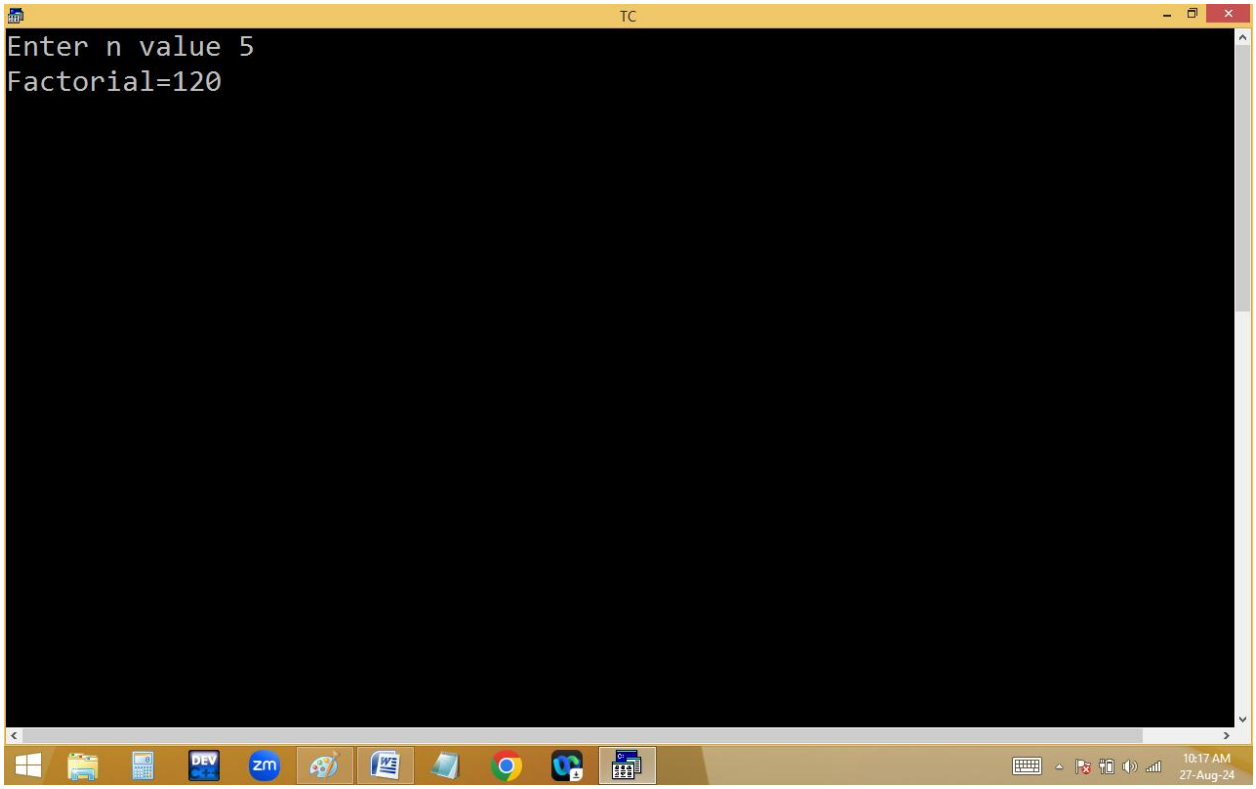

TC
Enter a, b values 3 3
Both are equal
Both are equal
10:08 AM
27-Aug-24

Finding factorial using pointer:



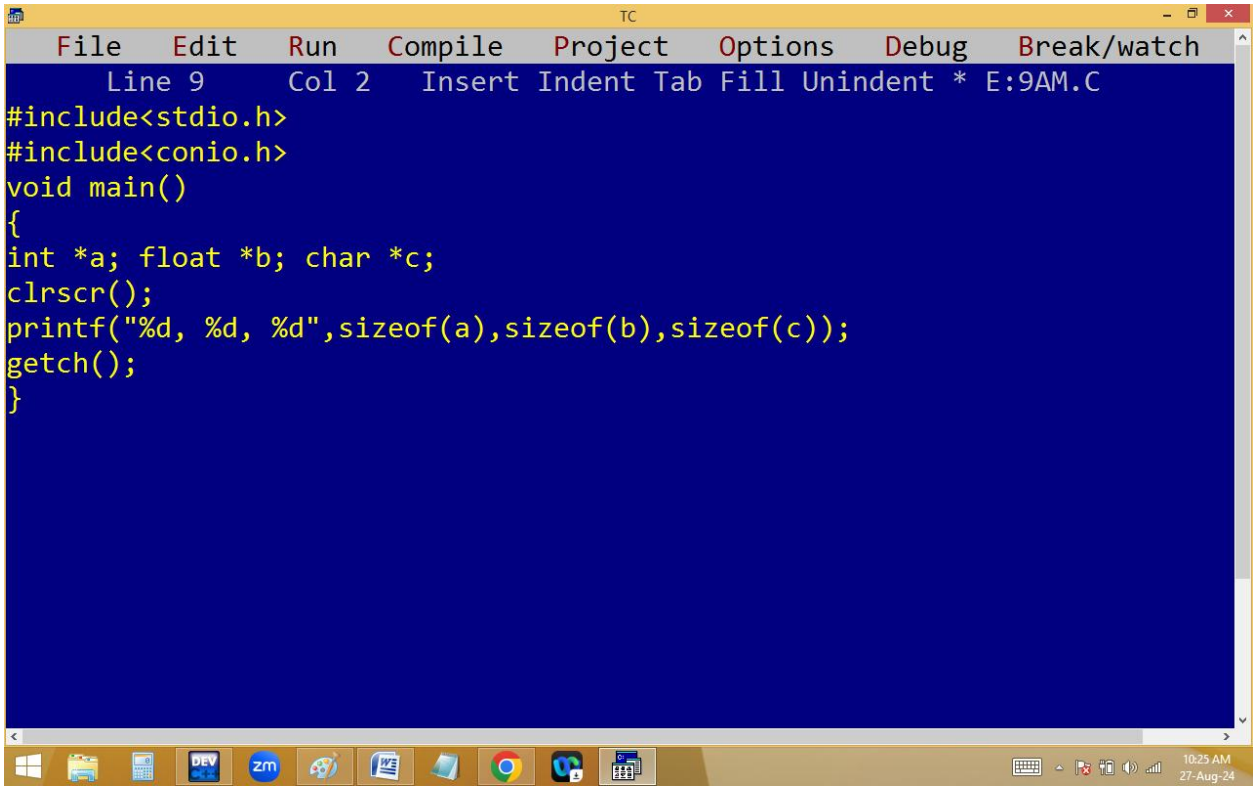


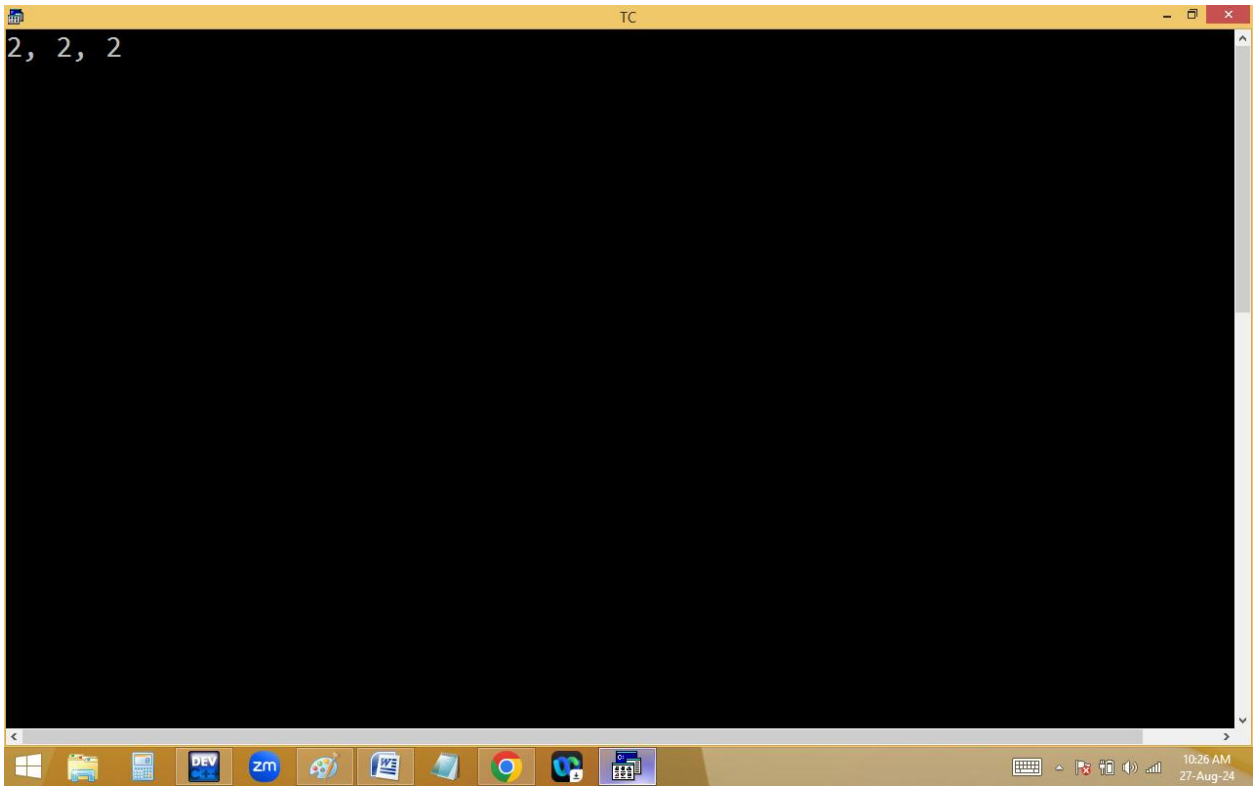

TC
File Edit Run Compile Project Options Debug Break/watch
Line 16 Col 37 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n, *p=&n,f=1;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(*p>1)
{
f=f* *p;
--*p; /* (*p)--; */
}
printf("Factorial=%d",f);
getch();
}
/* Always * gets least priority */_


TC
Enter n value 5
Factorial=120

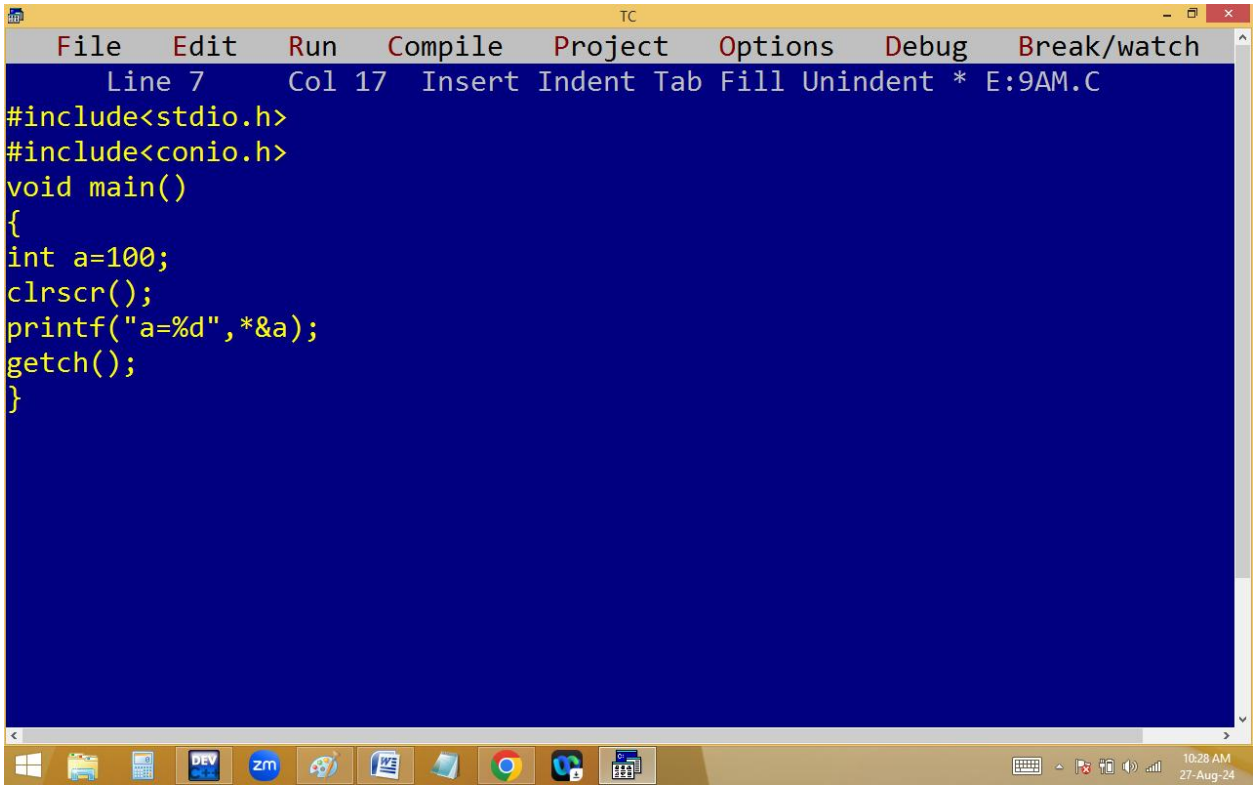
Finding pointer size:

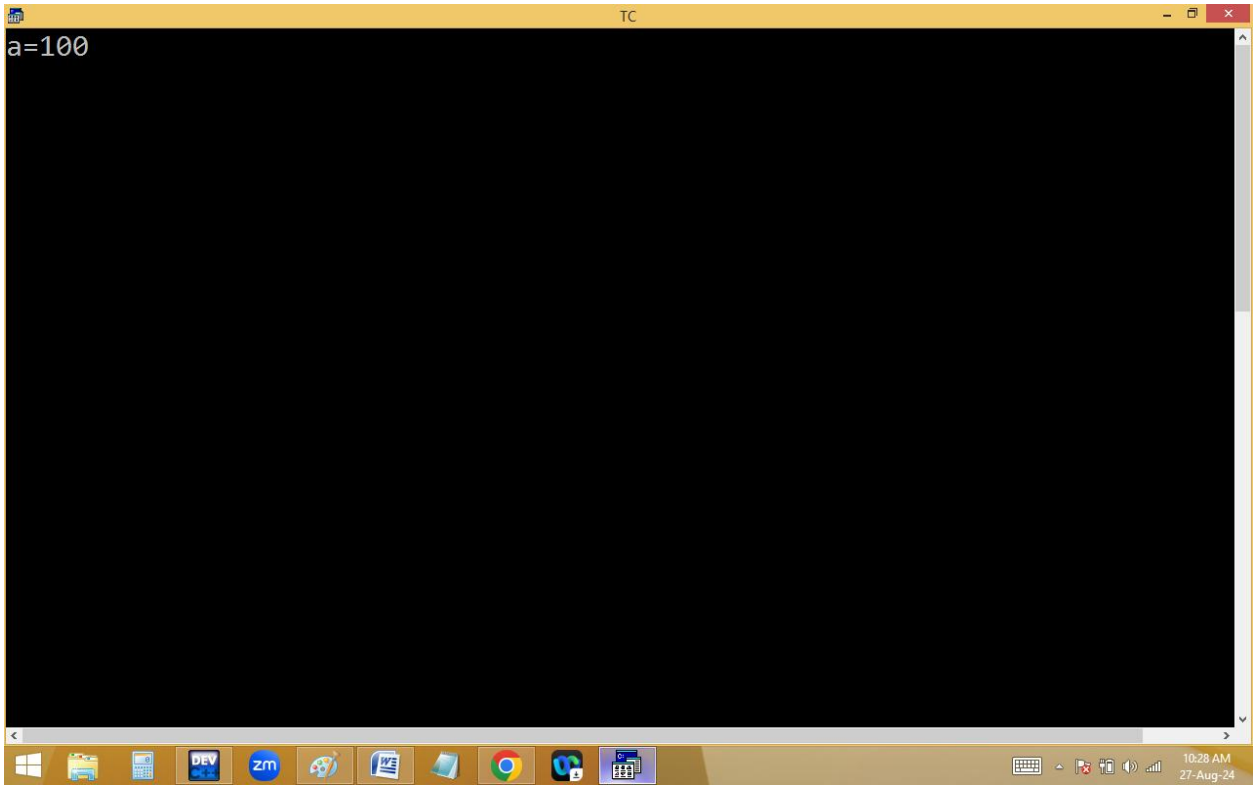
Pointer stores address of another variable and address is always unsigned int. due to this the pointer takes 2 / 4 / 8 bytes in 16 / 32 / 64 bit compilers.


TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 2 Insert Indent Tab Fill Unindent * E:9AM.C
#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();
}

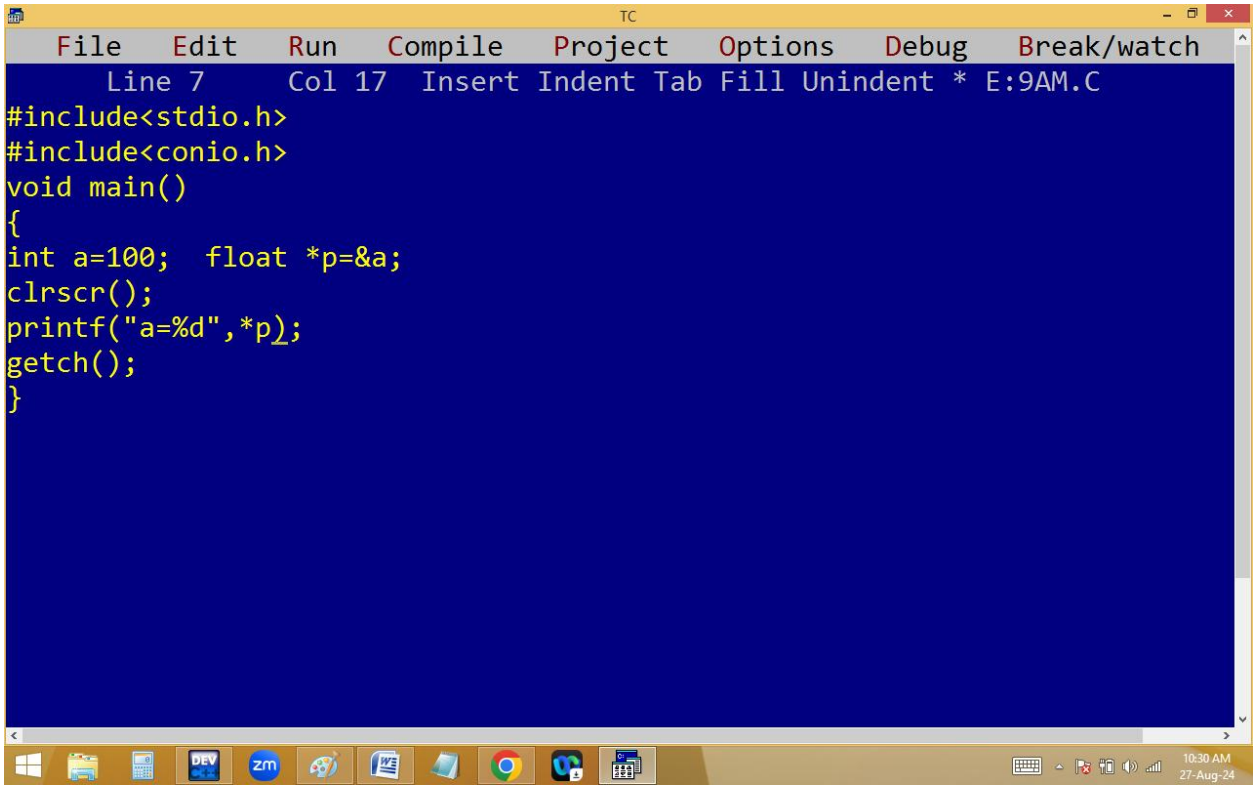

TC
2, 2, 2

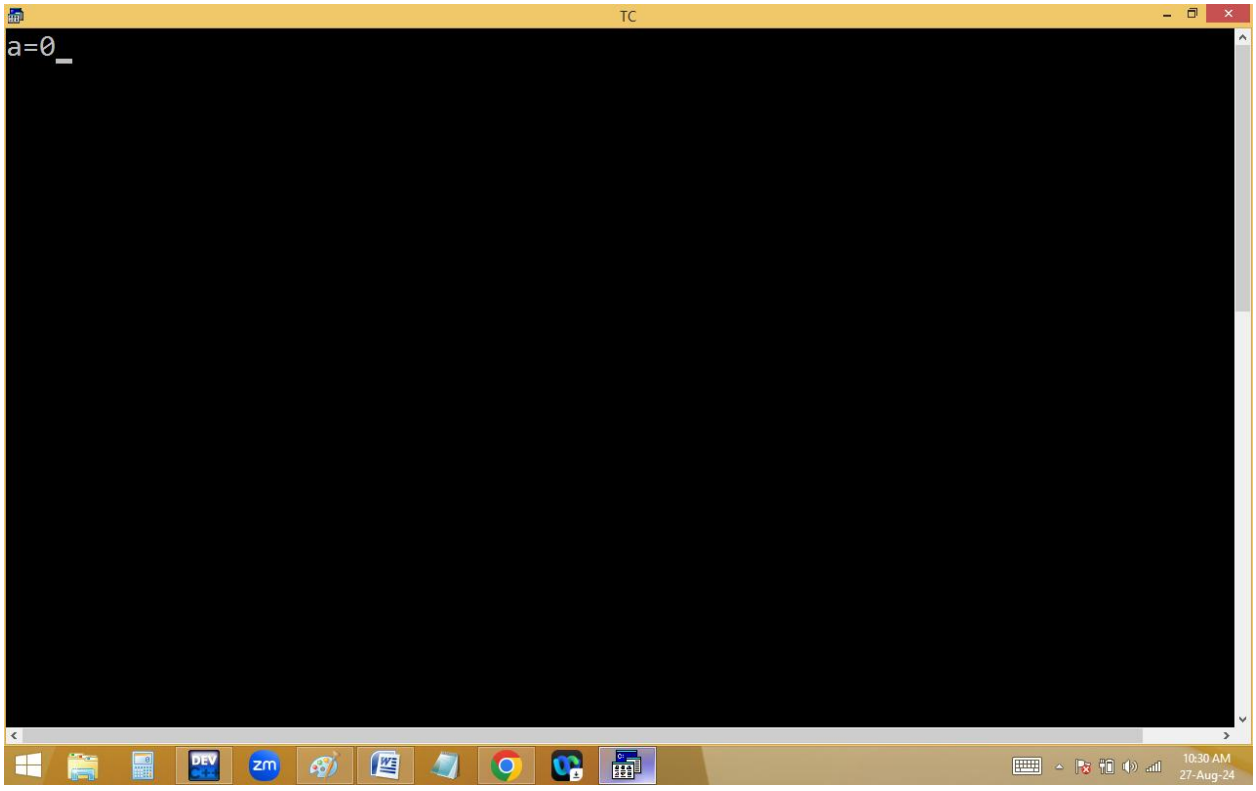
Finding a normal variable value using pointer mechanism:

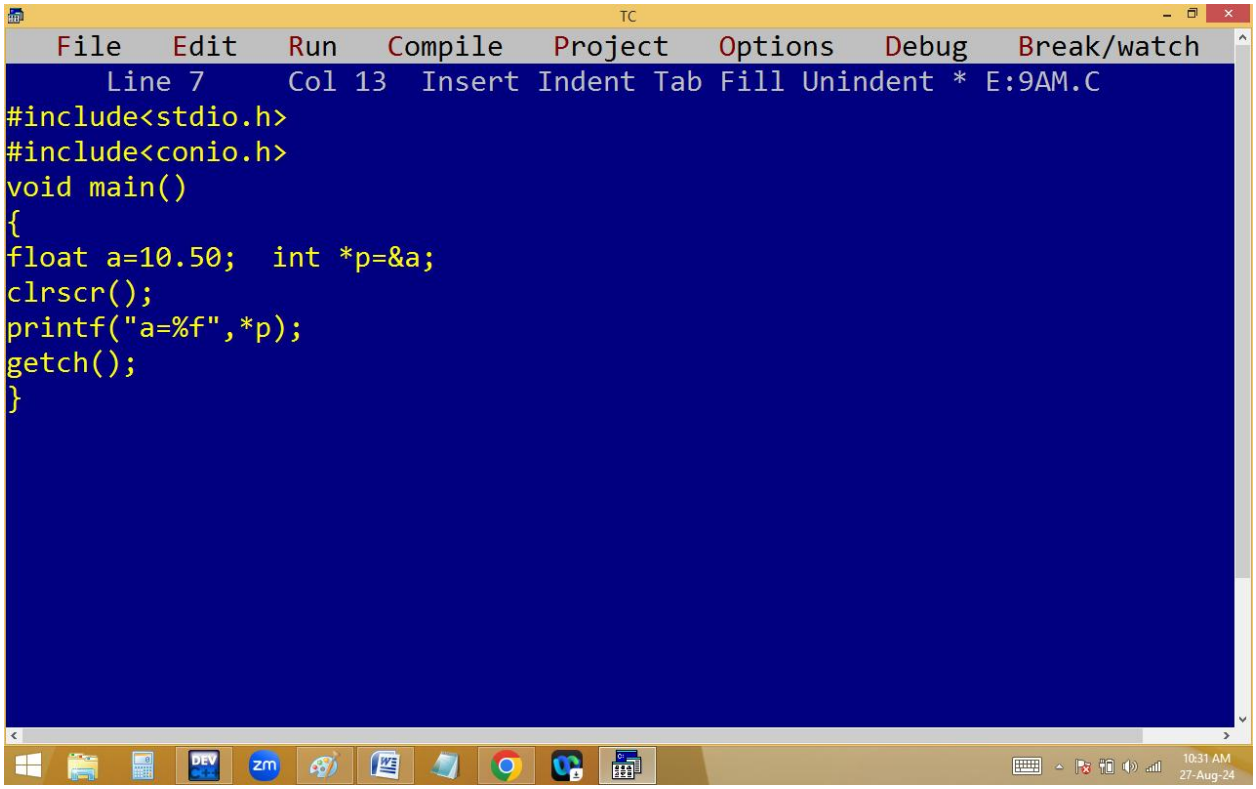


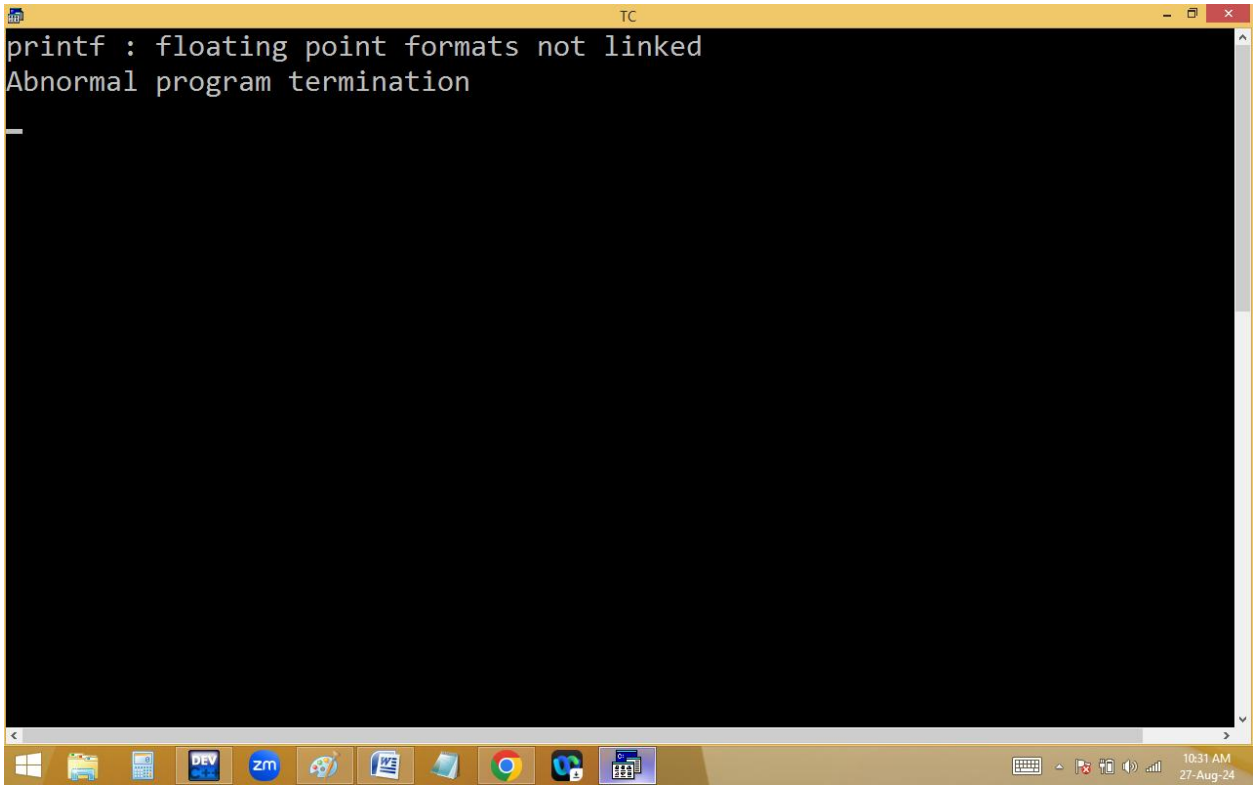


Pointer compatibility:



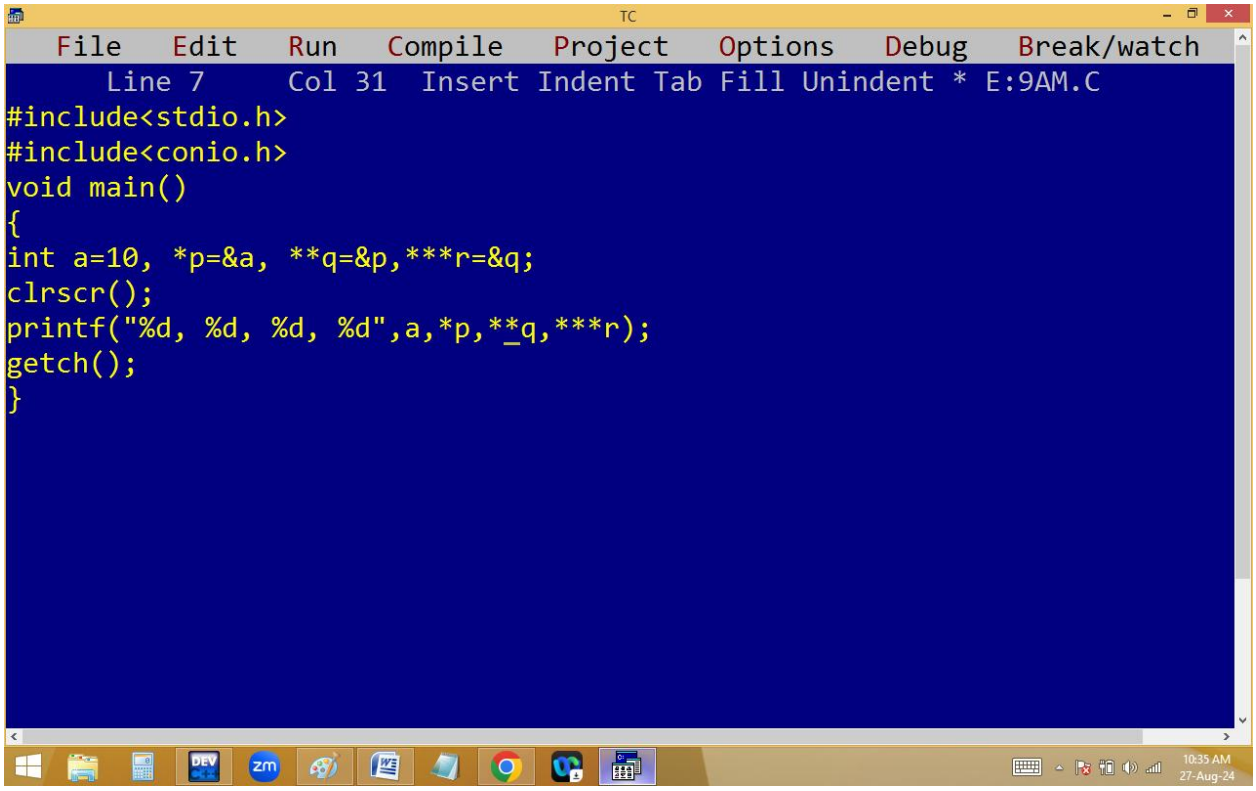


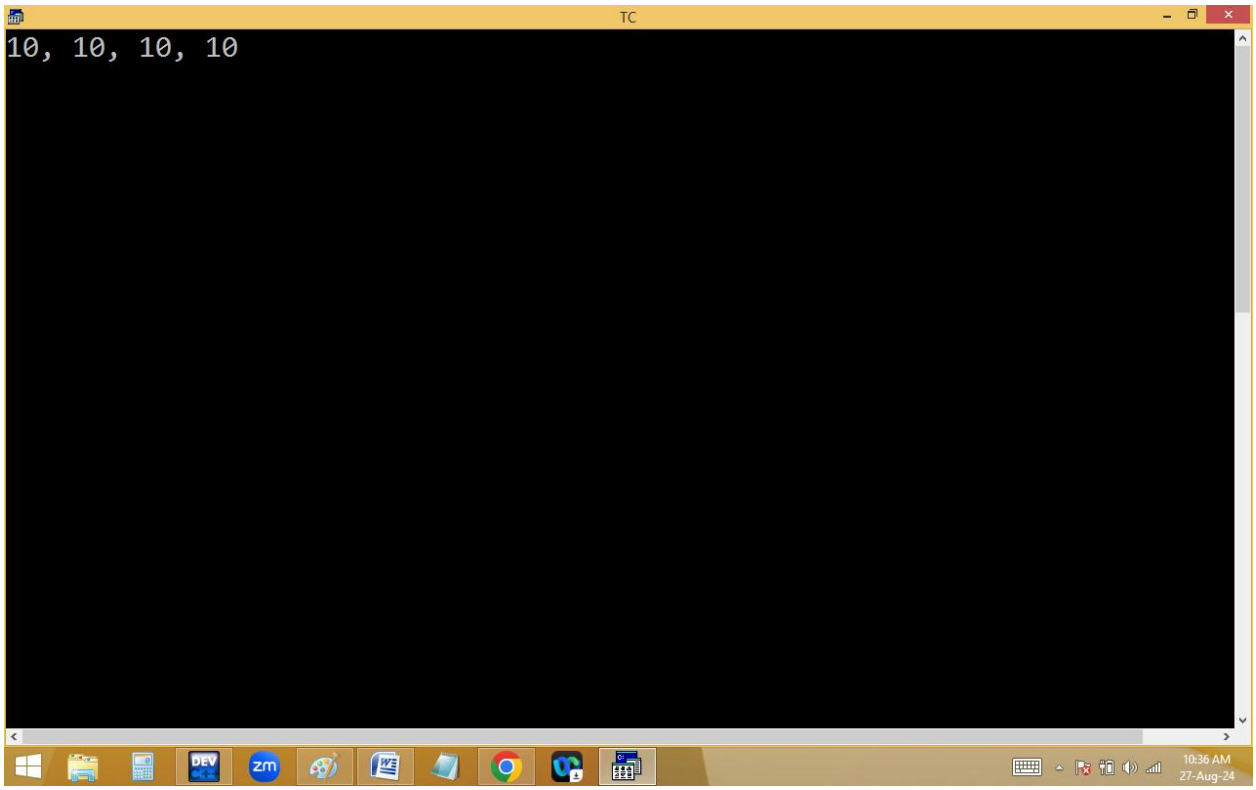

TC
File Edit Run Compile Project Options Debug Break/watch
Line 7 Col 13 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
float a=10.50; int *p=&a;
clrscr();
printf("a=%f",*p);
getch();
}

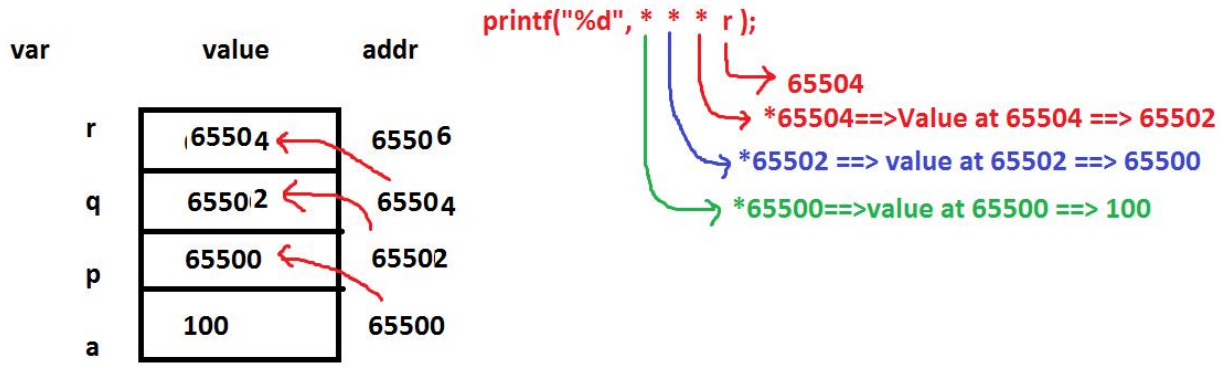

TC
printf : floating point formats not linked
Abnormal program termination

Double pointer / pointer to pointer:

The pointer which stores the address of another pointer is called double pointer and they are used to handle dynamic multi dimensional array.


TC
File Edit Run Compile Project Options Debug Break/watch
Line 7 Col 31 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a, **q=&p,***r=&q;
clrscr();
printf("%d, %d, %d, %d",a,*p,**q,***r);
getch();
}


TC
10, 10, 10, 10


var
value
addr
r
65504
65506
q
65502
65504
p
65500
65502
a
100
65500
printf("%d", * * * r);
65504
*65504==>Value at 65504 ==> 65502
*65502 ==> value at 65502 ==> 65500
*65500==>value at 65500 ==> 100