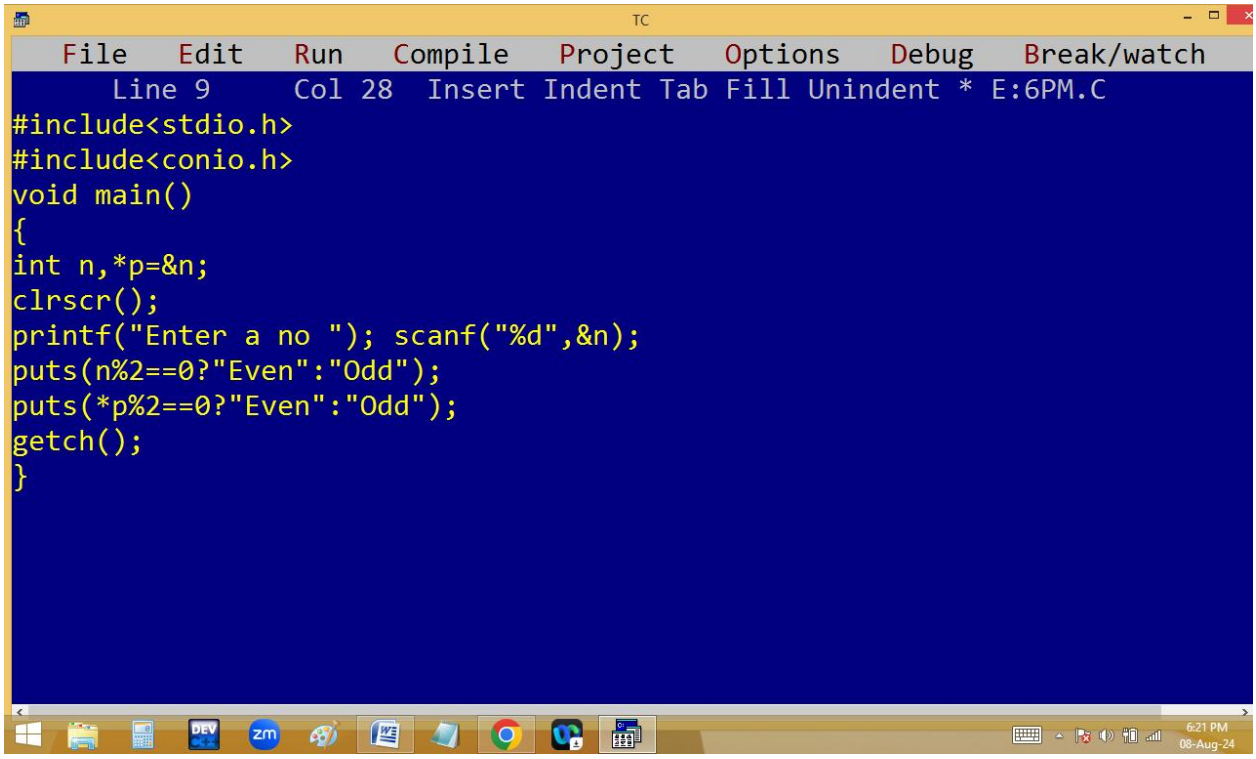
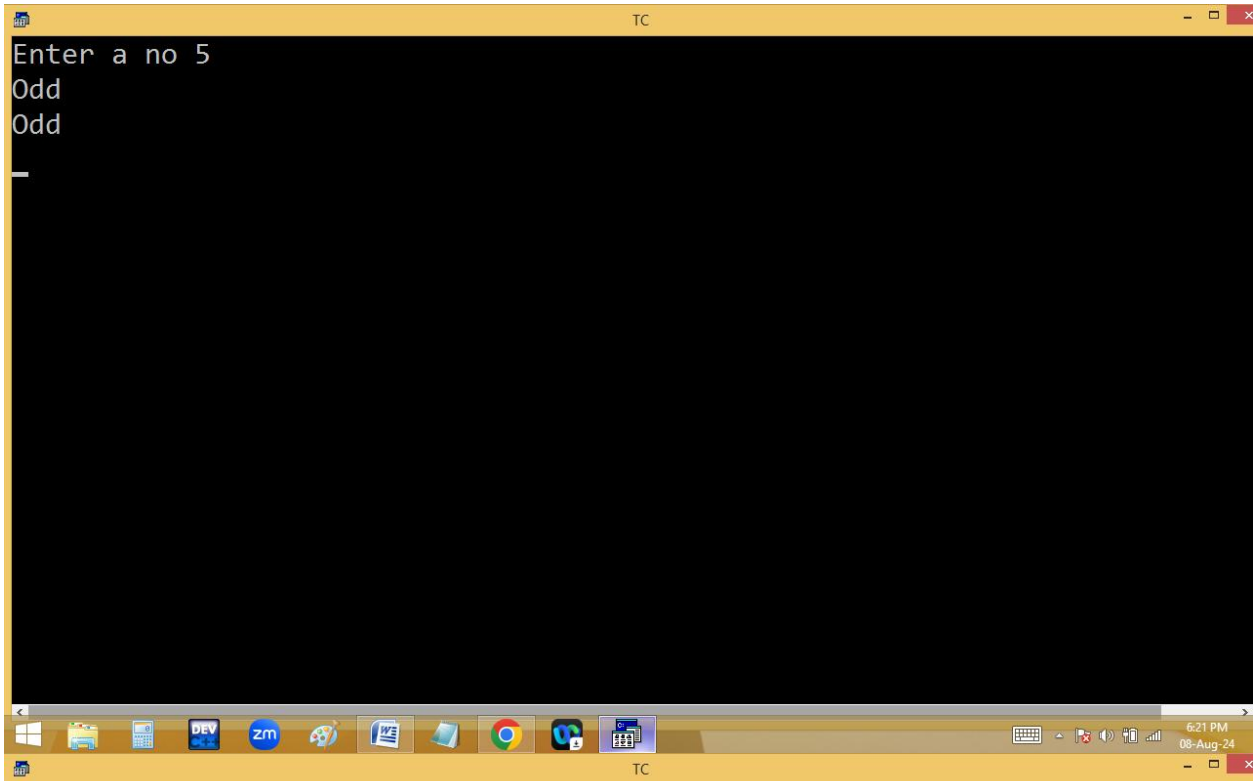
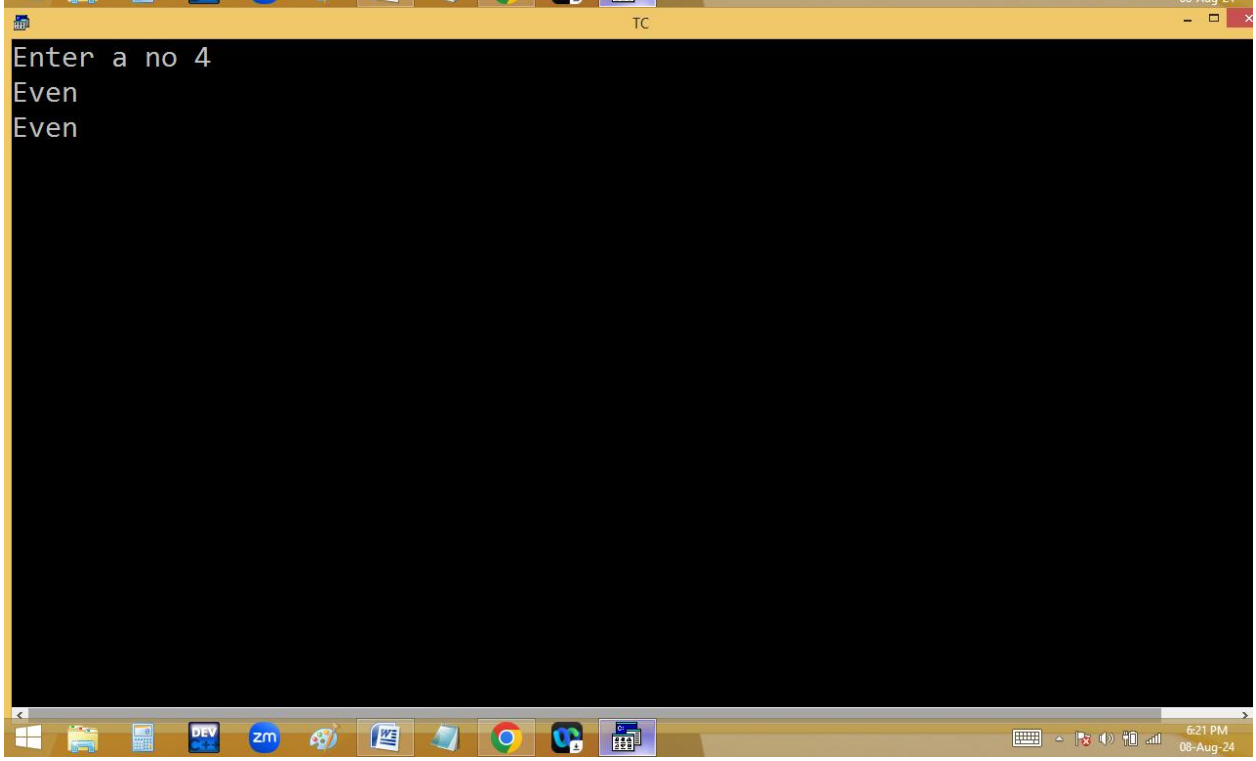
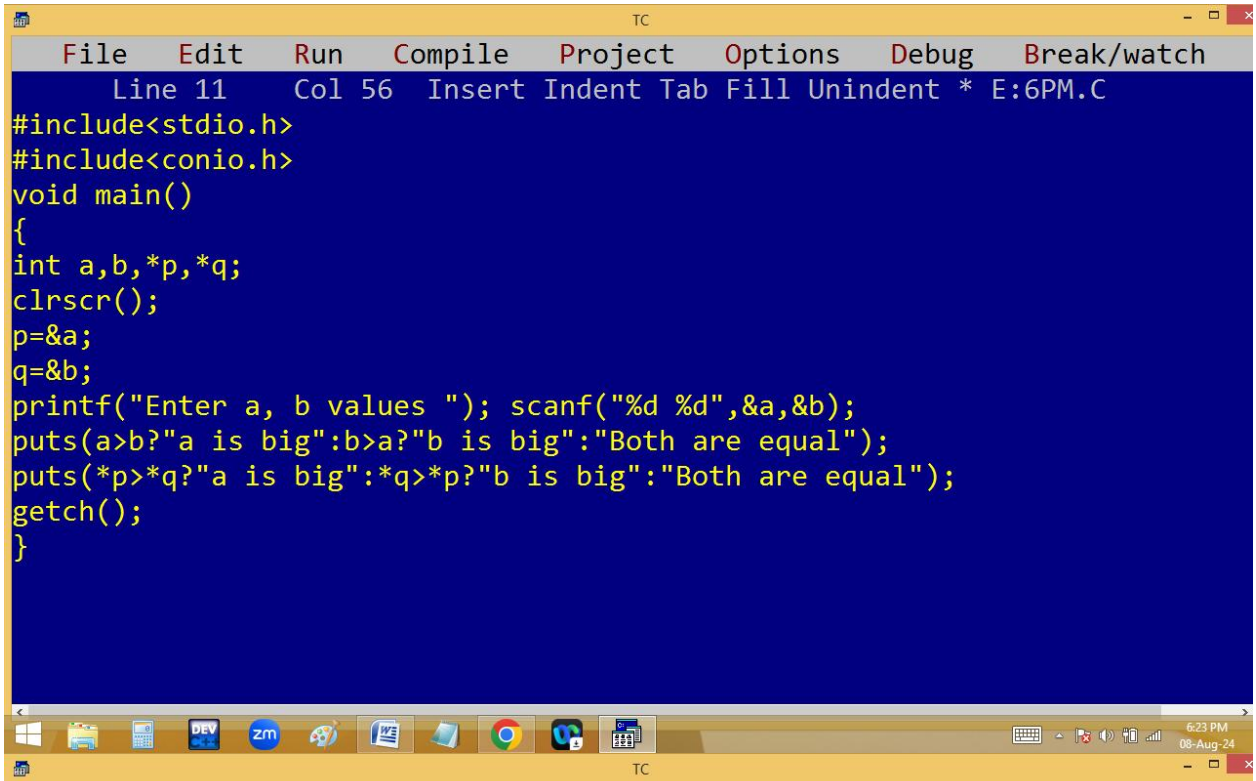
Finding even/odd using pointer:

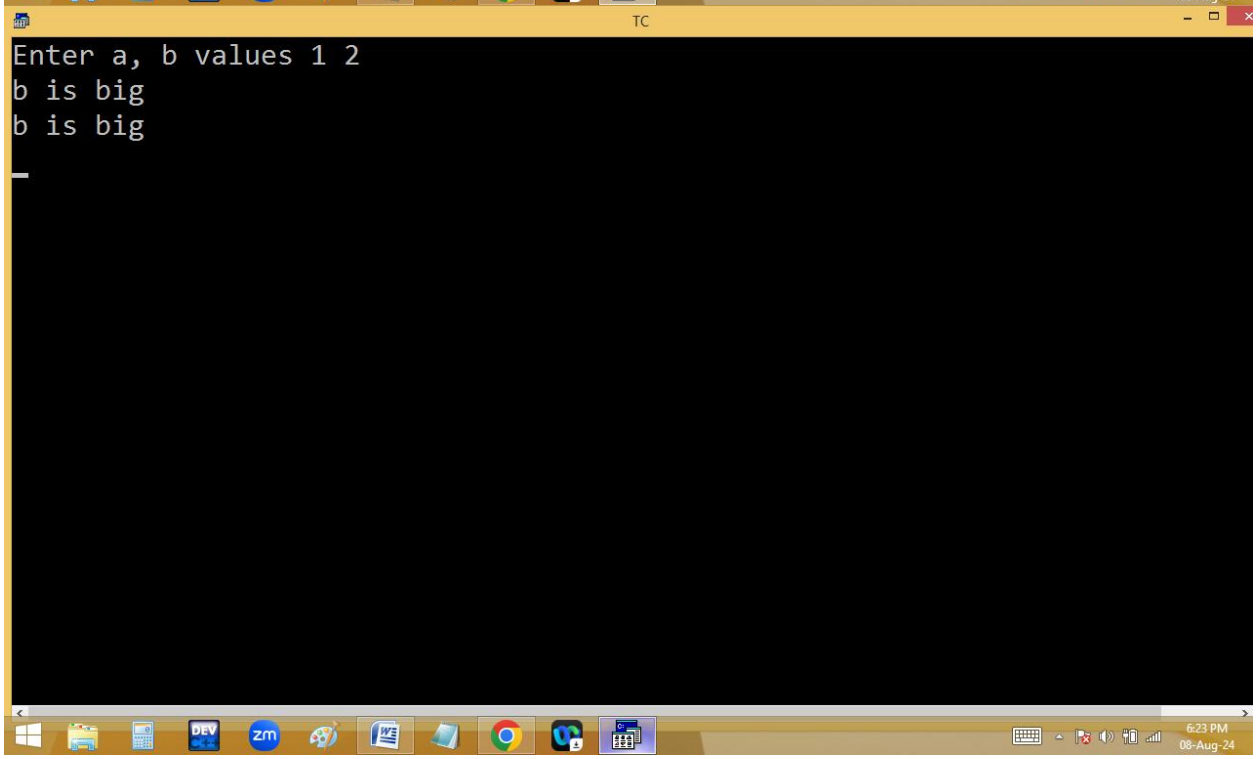




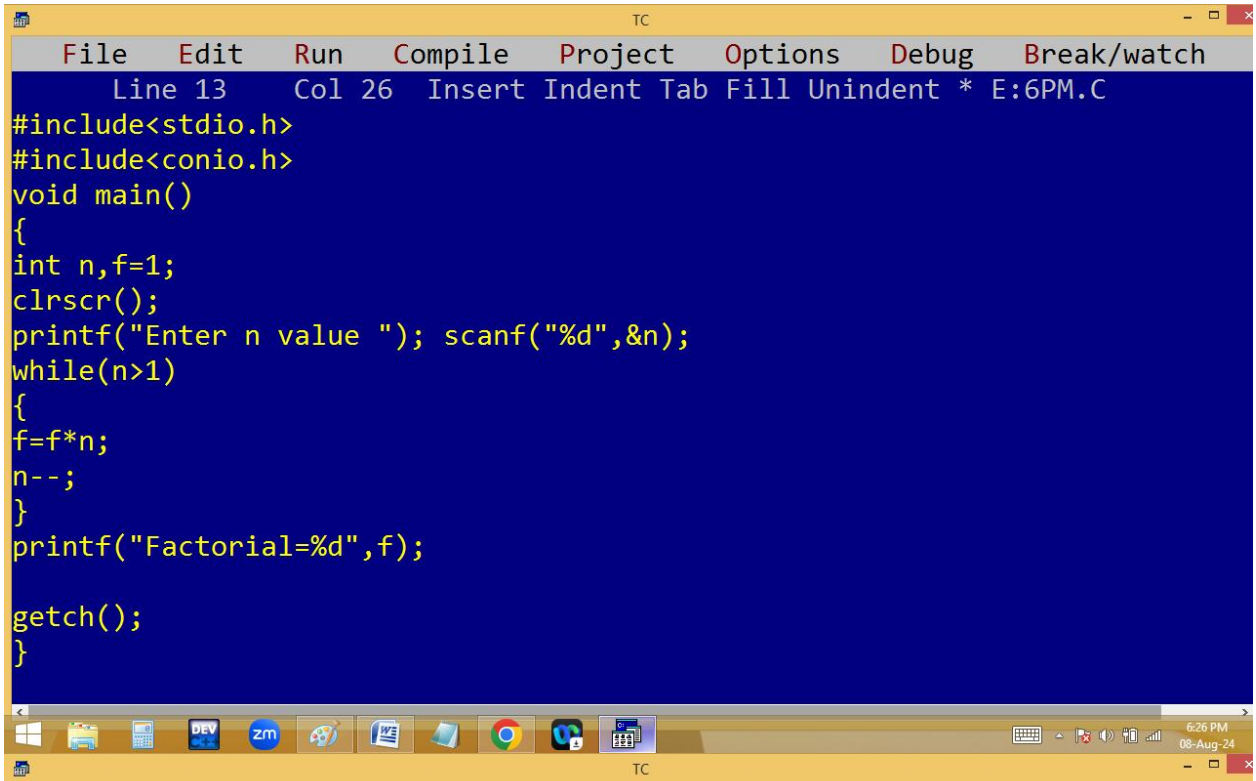


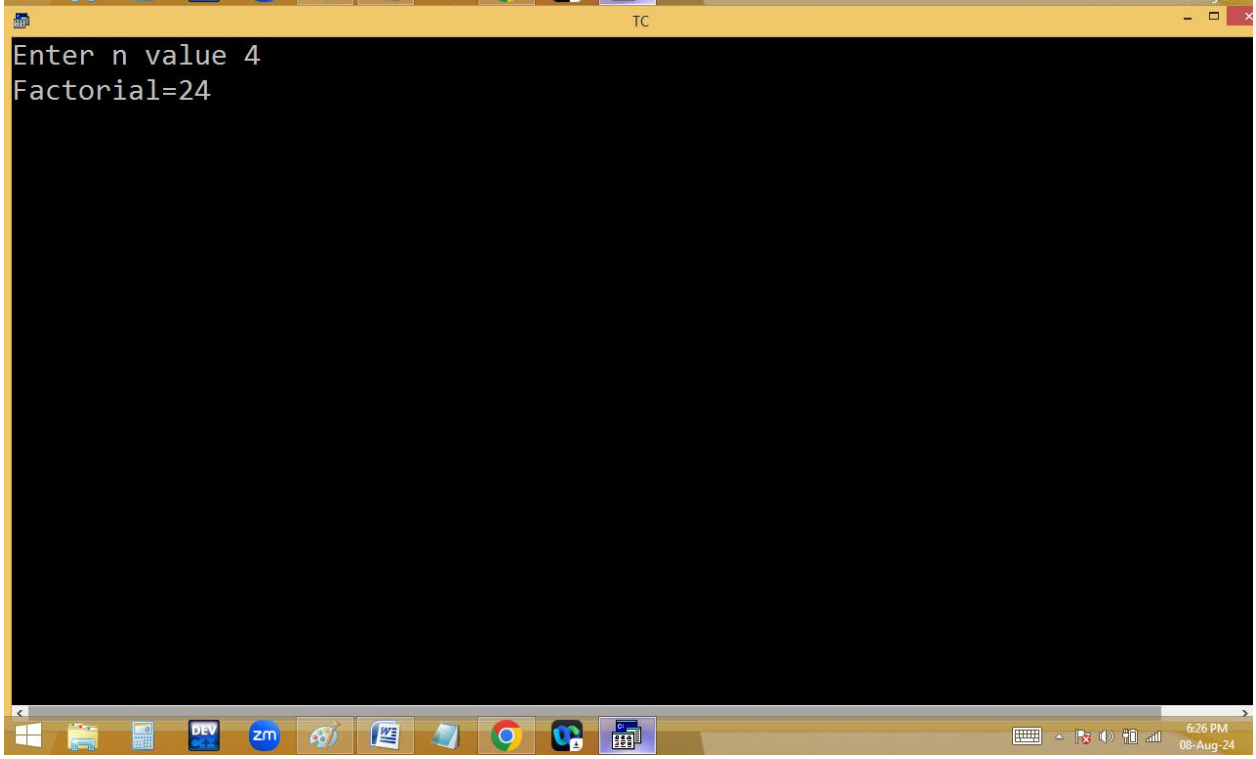
Finding max in 2 no's using pointer:

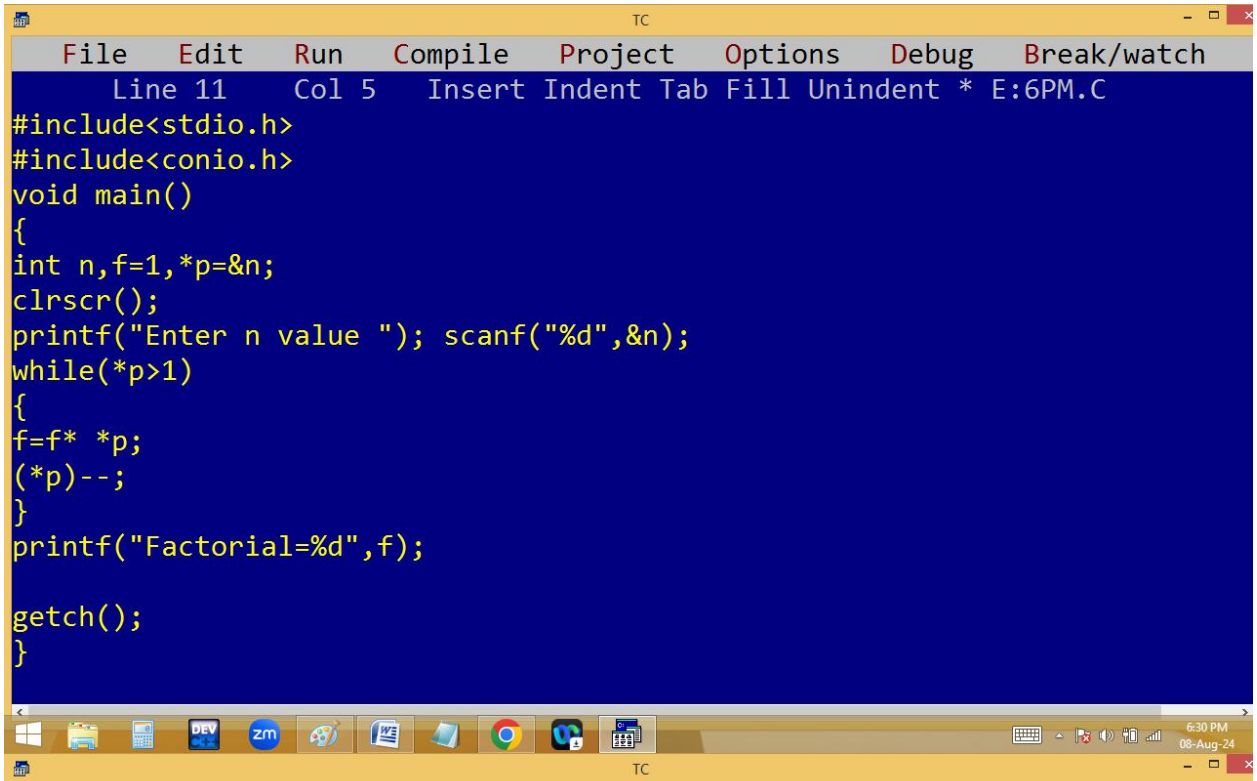

TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 56 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,*p,*q;
clrscr();
p=&a;
q=&b;
printf("Enter a, b values "); scanf("%d %d",&a,&b);
puts(a>b?"a is big":b>a?"b is big":"Both are equal");
puts(*p>*q?"a is big":*q>*p?"b is big":"Both are equal");
getch();
}

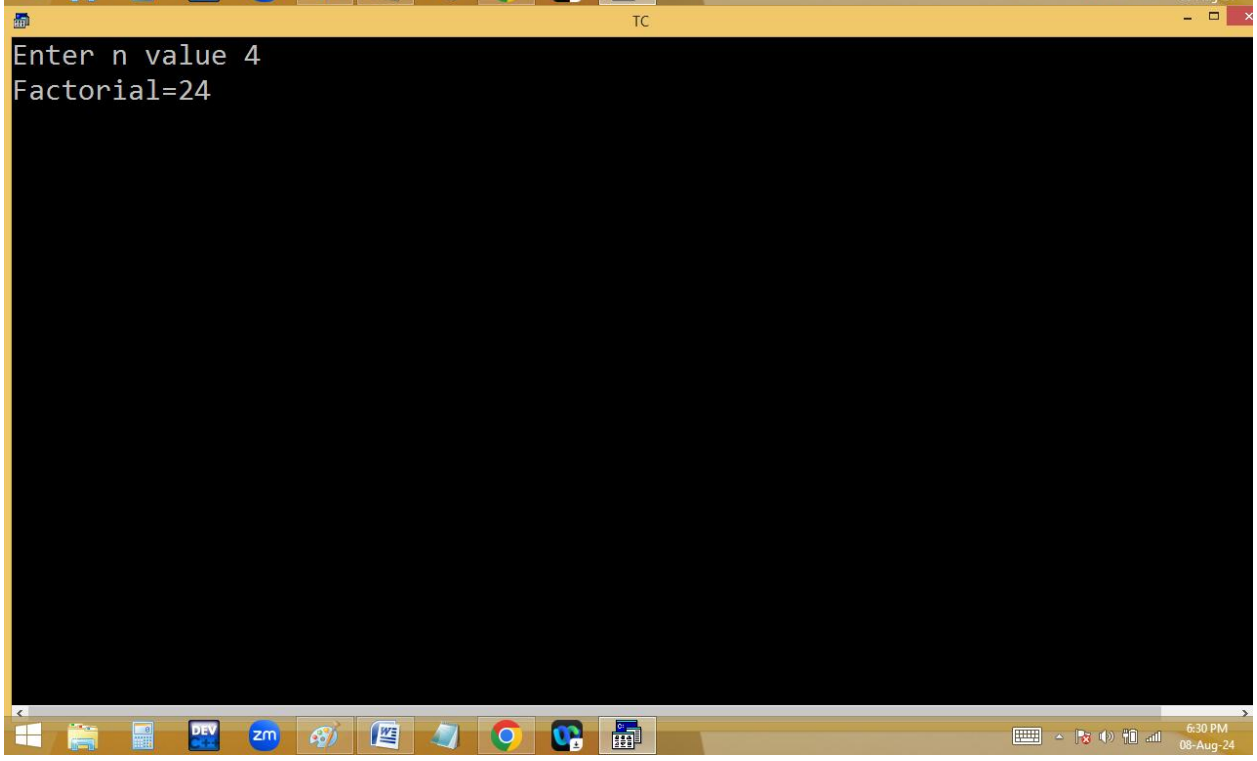

TC
Enter a, b values 1 2
b is big
b is big

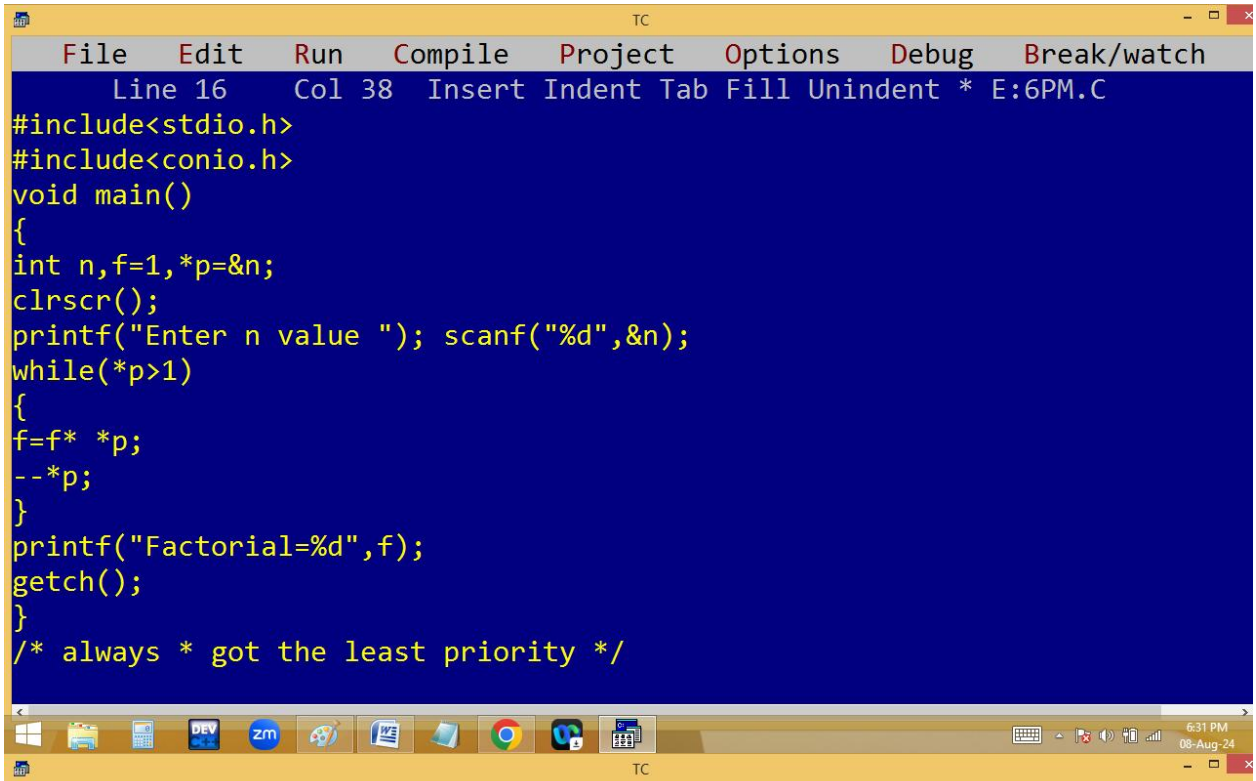
Finding factorial using pointer:

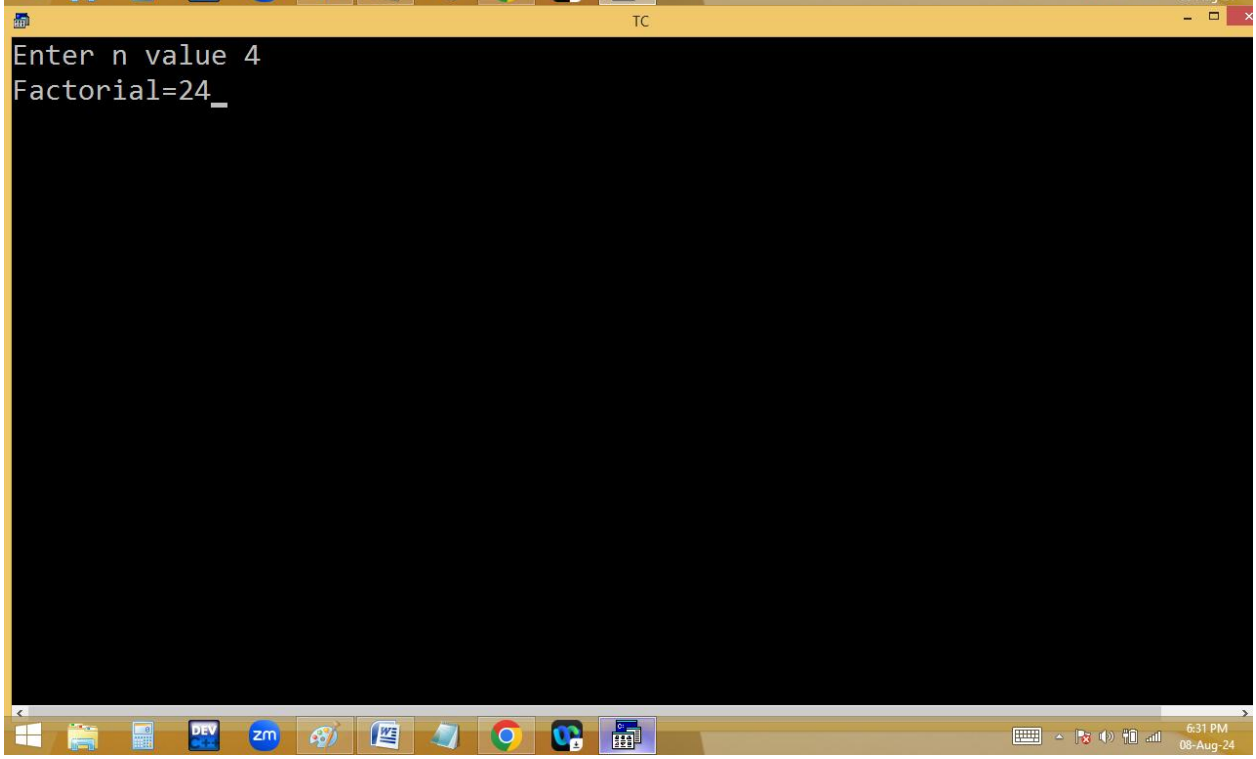

TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 26 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,f=1;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(n>1)
{
f=f*n;
n--;
}
printf("Factorial=%d",f);
getch();
}


TC
Enter n value 4
Factorial=24

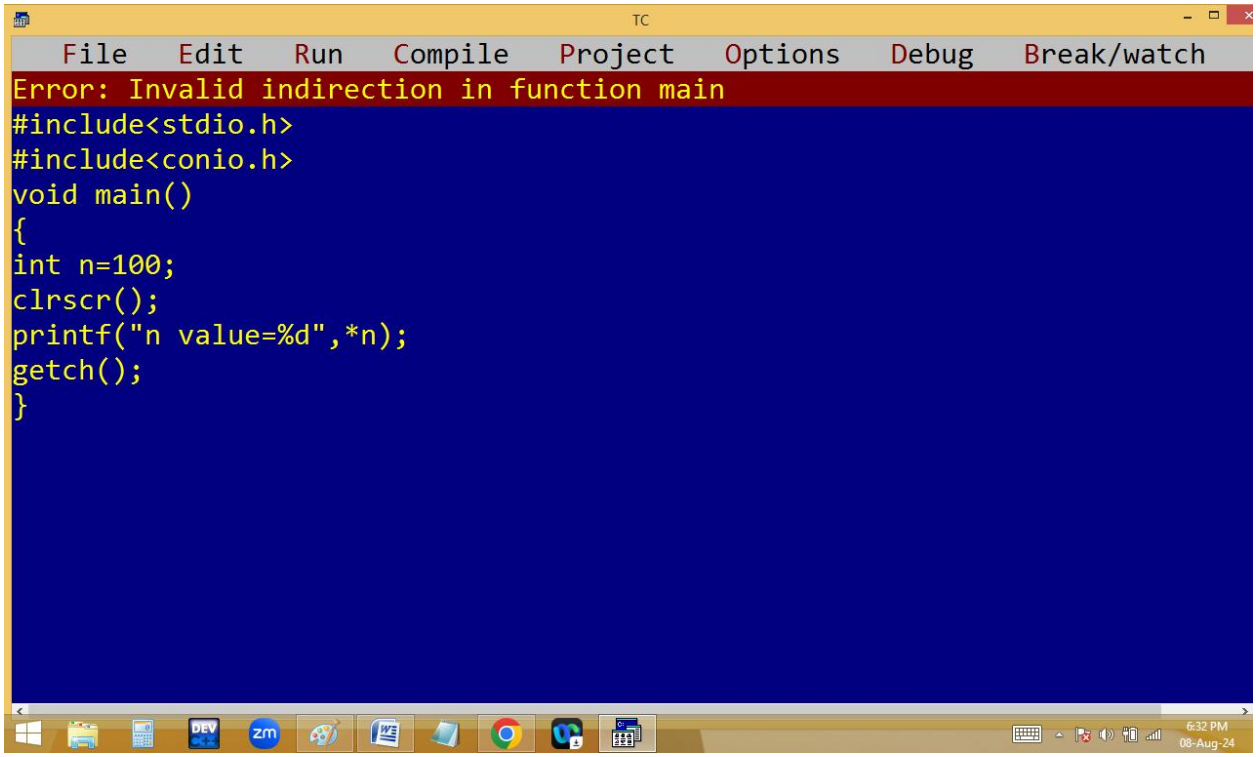

TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 5 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,f=1,*p=&n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(*p>1)
{
f=f* *p;
(*p)--;
}
printf("Factorial=%d",f);
getch();
}

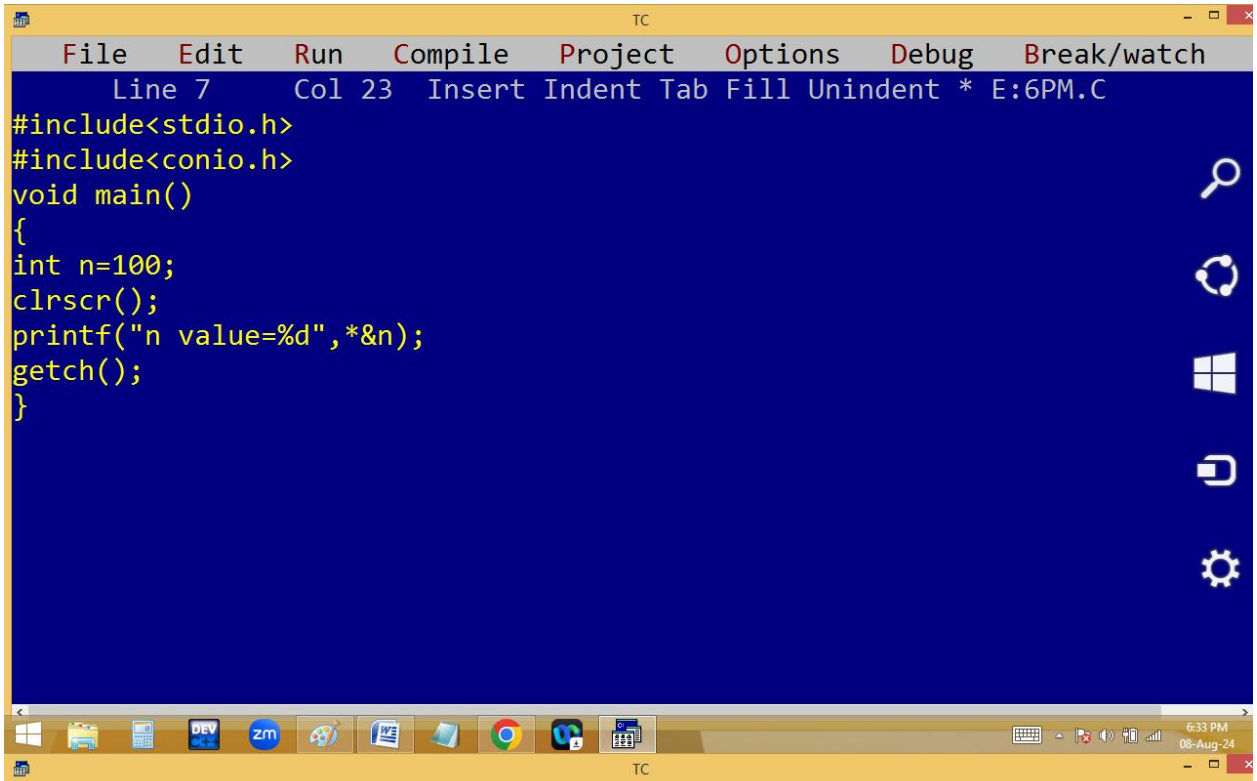

TC
Enter n value 4
Factorial=24

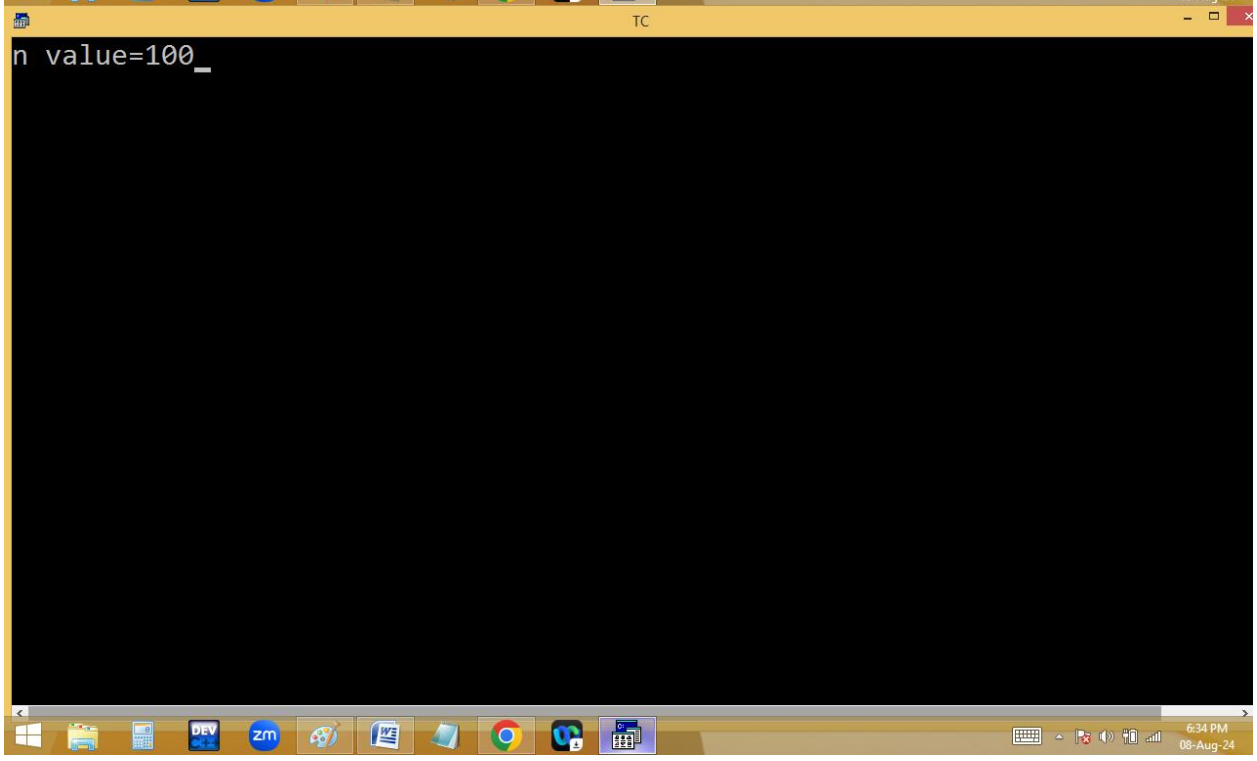

TC
File Edit Run Compile Project Options Debug Break/watch
Line 16 Col 38 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,f=1,*p=&n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
while(*p>1)
{
f=f* *p;
--*p;
}
printf("Factorial=%d",f);
getch();
}
/* always * got the least priority */


TC
Enter n value 4
Factorial=24

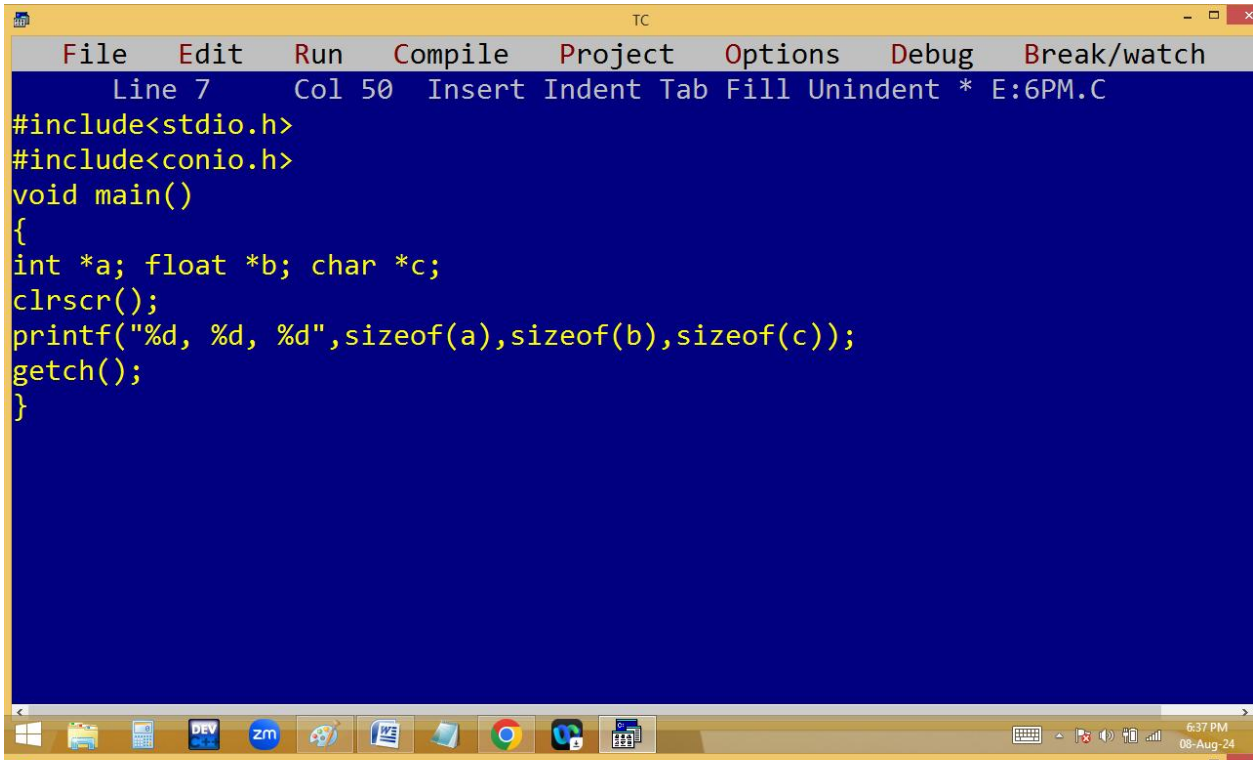
Finding a normal variable value using pointer mechanism:

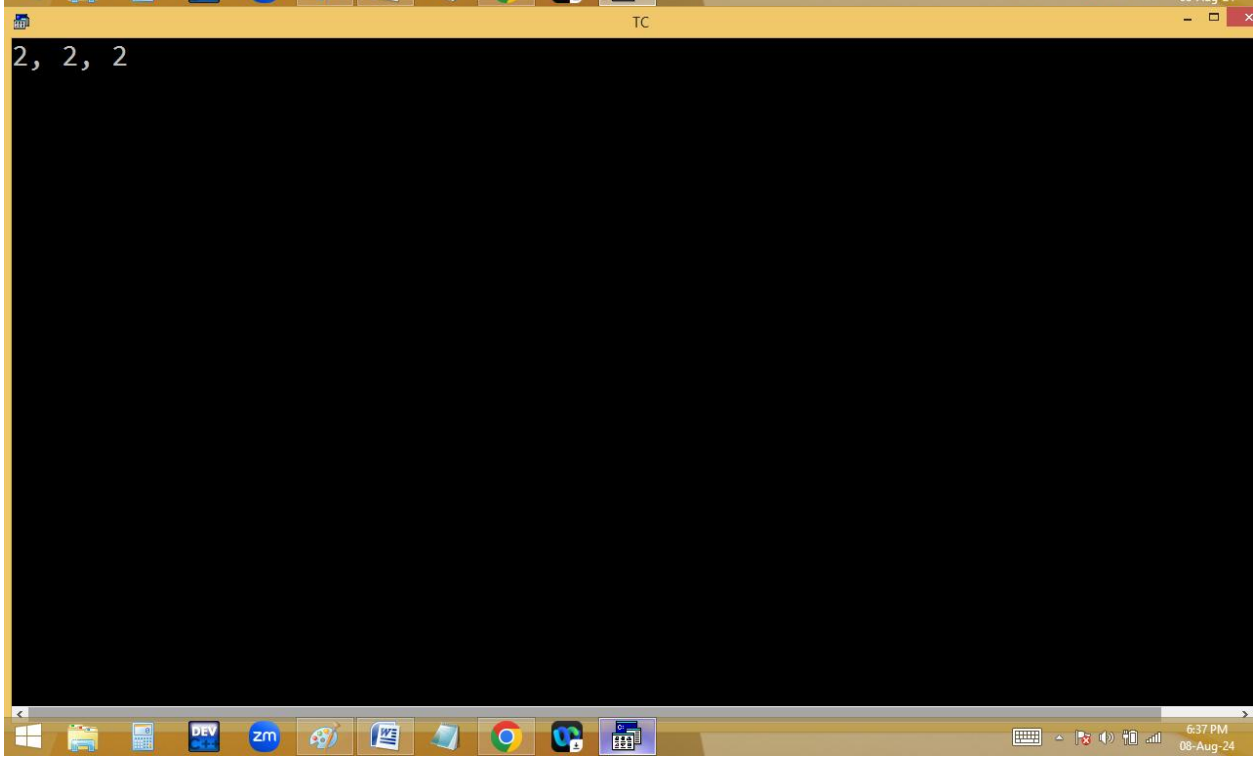



TC
File Edit Run Compile Project Options Debug Break/watch
Line 7 Col 23 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n=100;
clrscr();
printf("n value=%d",*&n);
getch();
}
6:33 PM
08-Aug-24

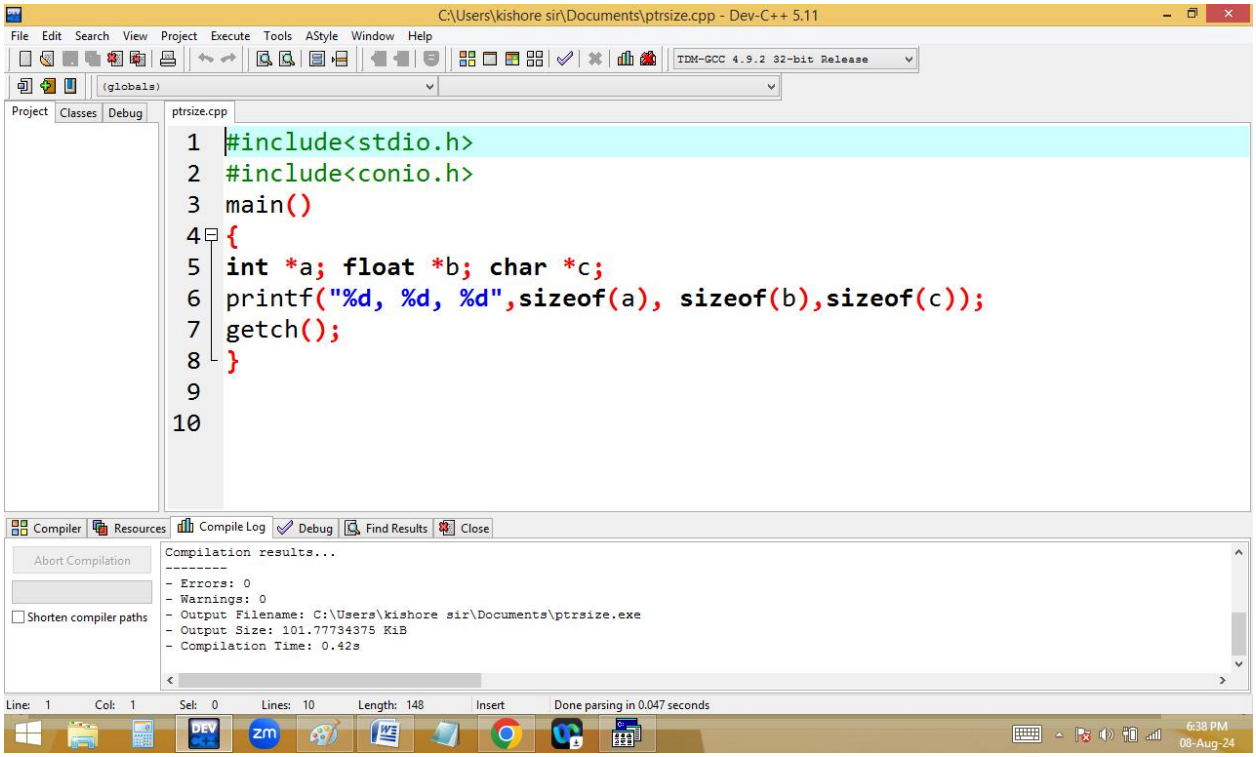

TC
n value=100
6:34 PM
08-Aug-24

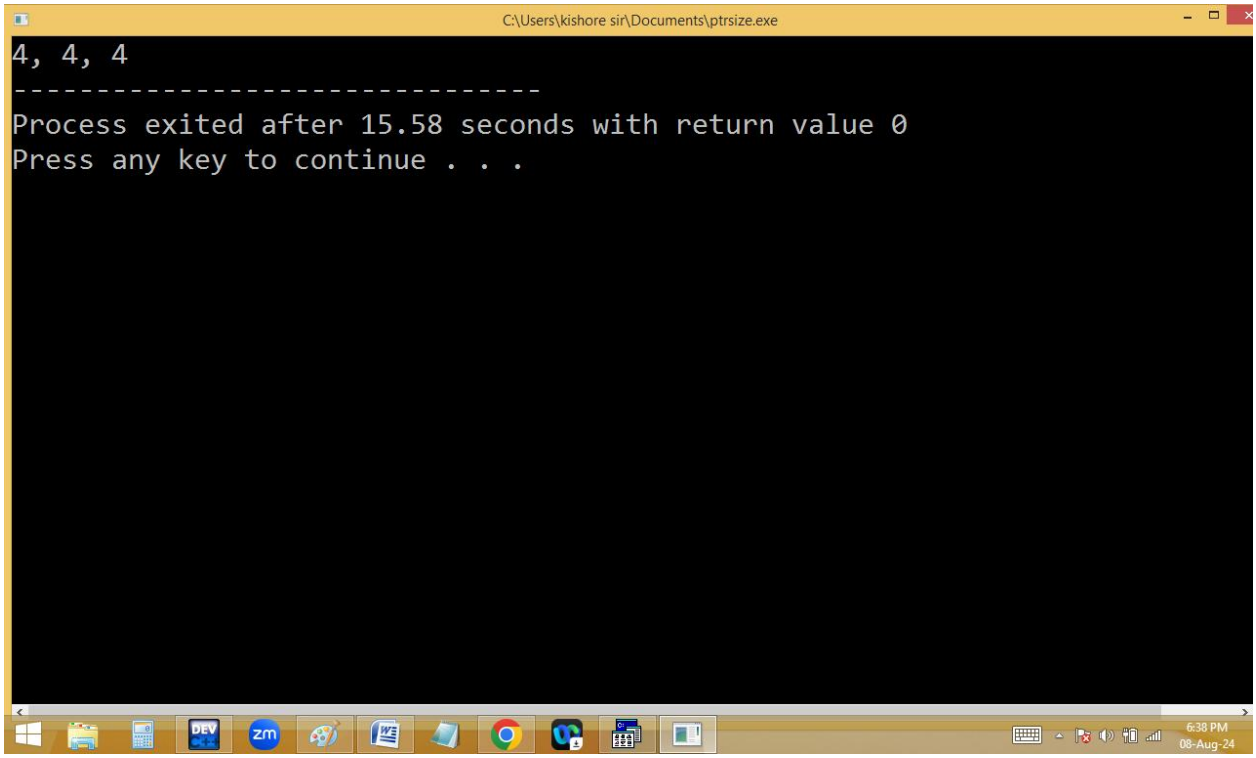
Finding pointer size: Any type of pointer it stores only the base address and address always unsigned int. Due to this pointer take 2 / 4 / 8 bytes in 16 / 32 / 64 bit compilers.



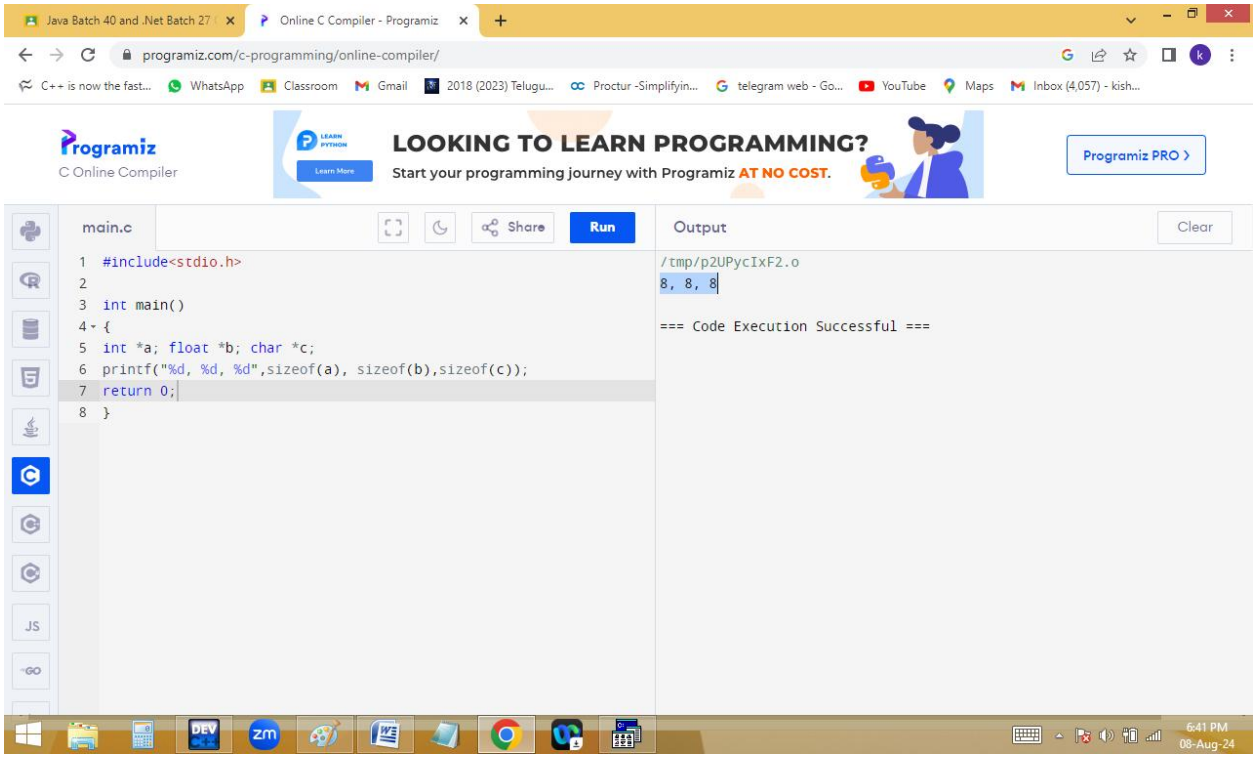


In dev c++:

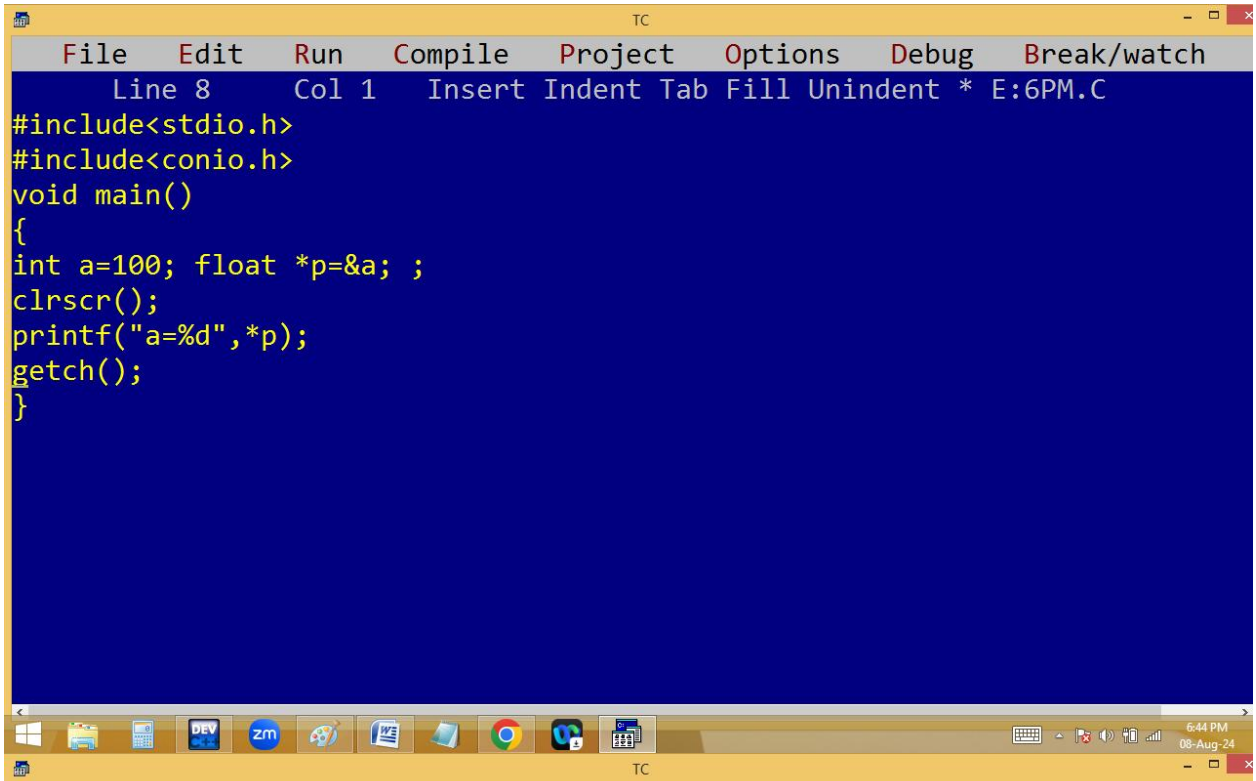


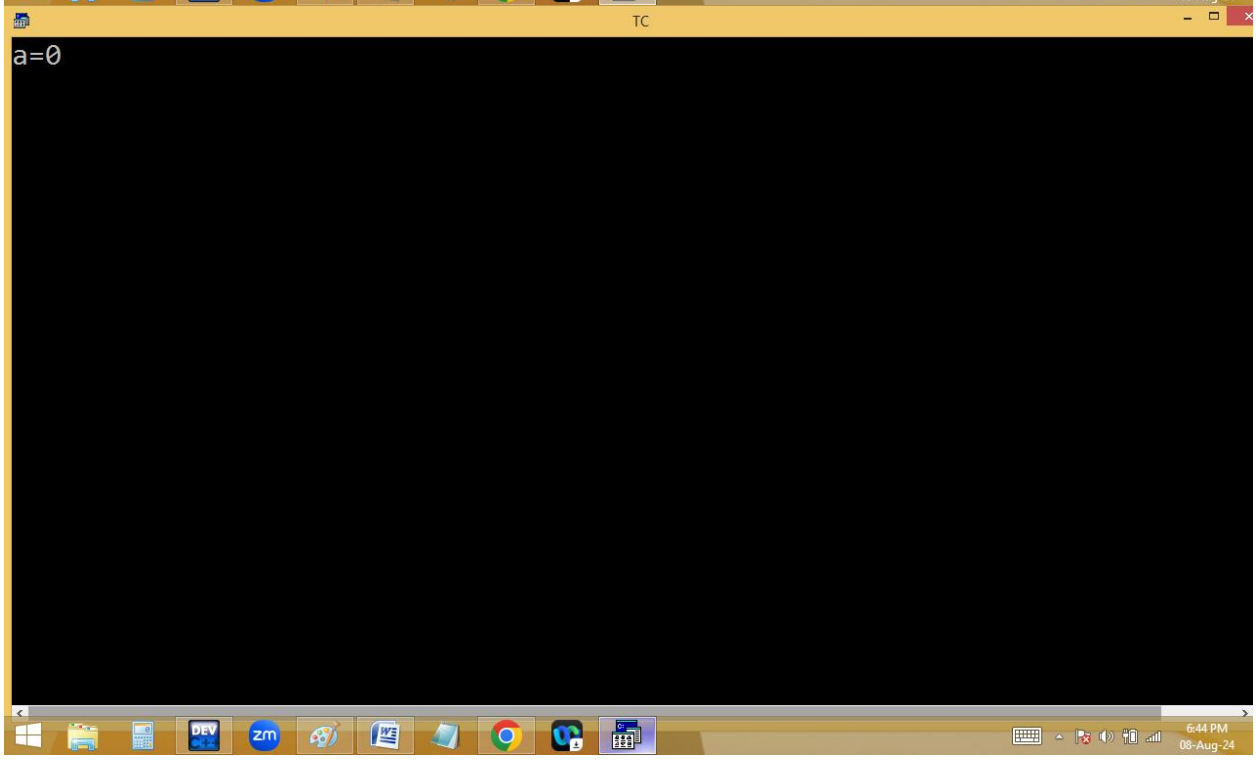


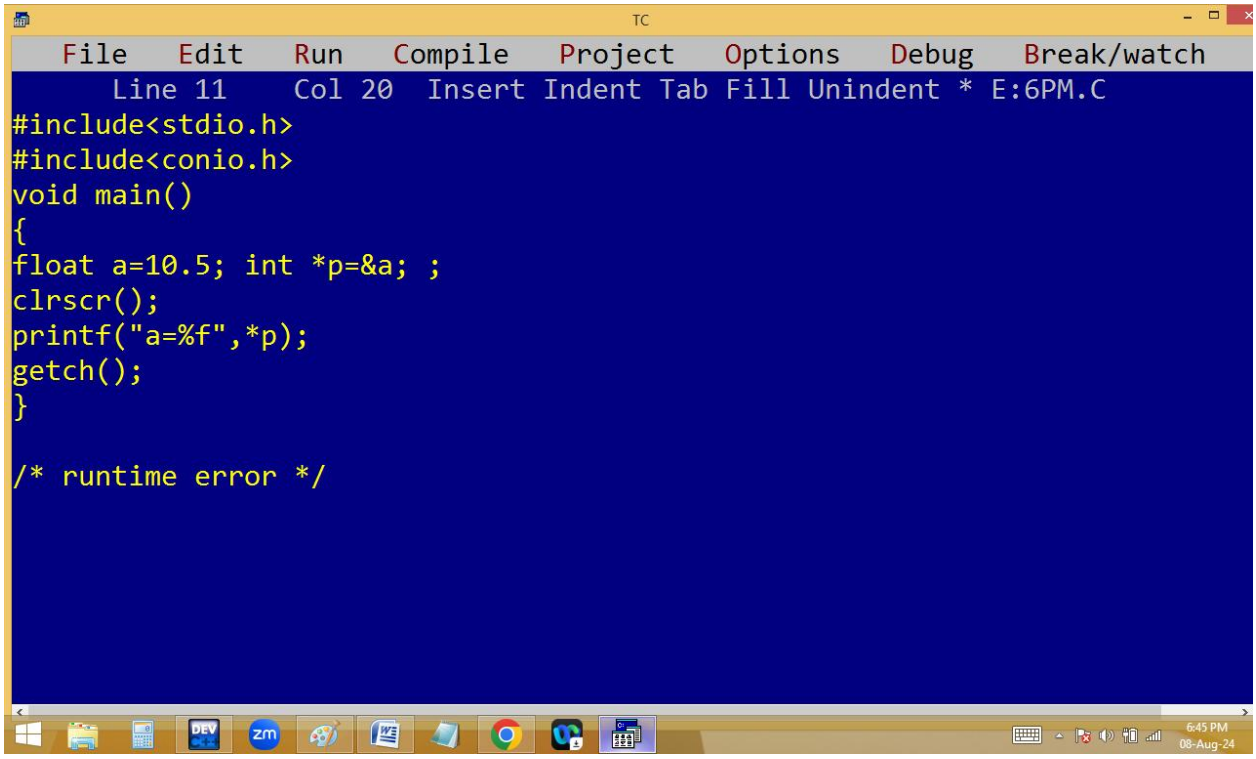
In online compiler:



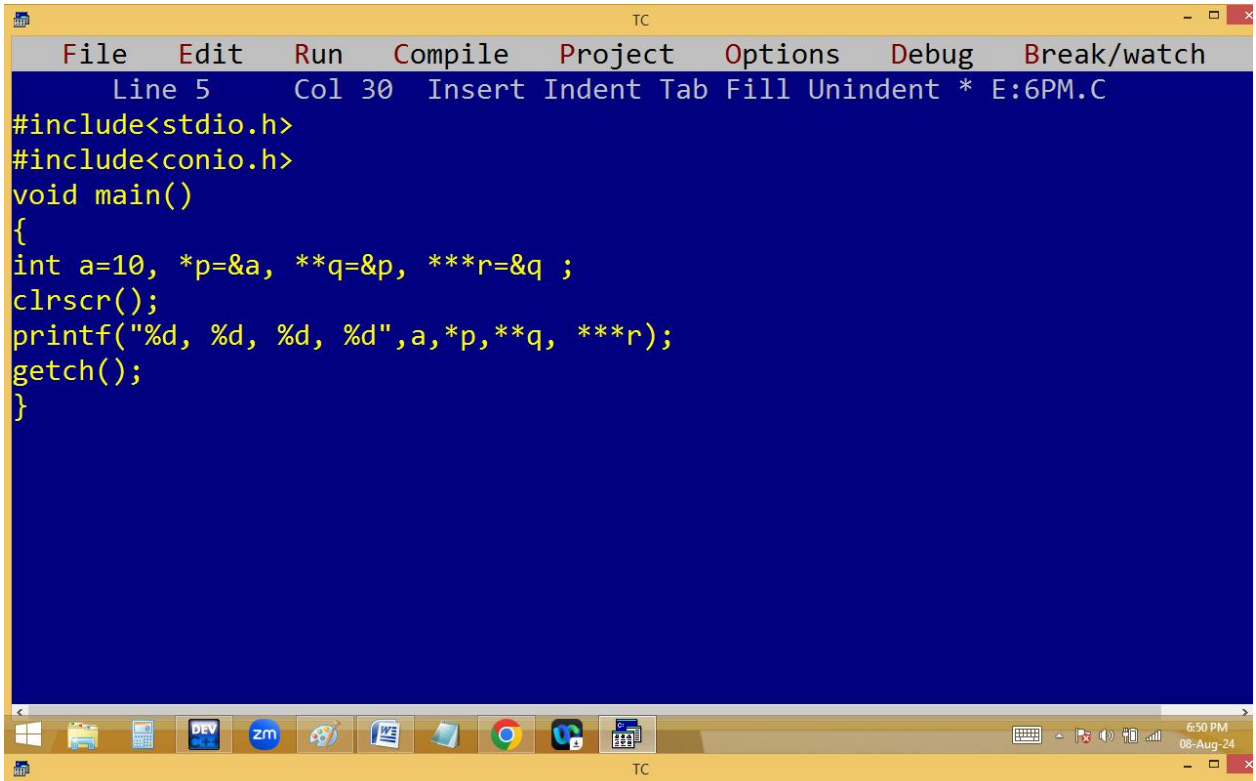
Pointer compatibility: pointer stores only the same type of variable address. When we are trying to store different type of address, it gives garbage or runtime error.

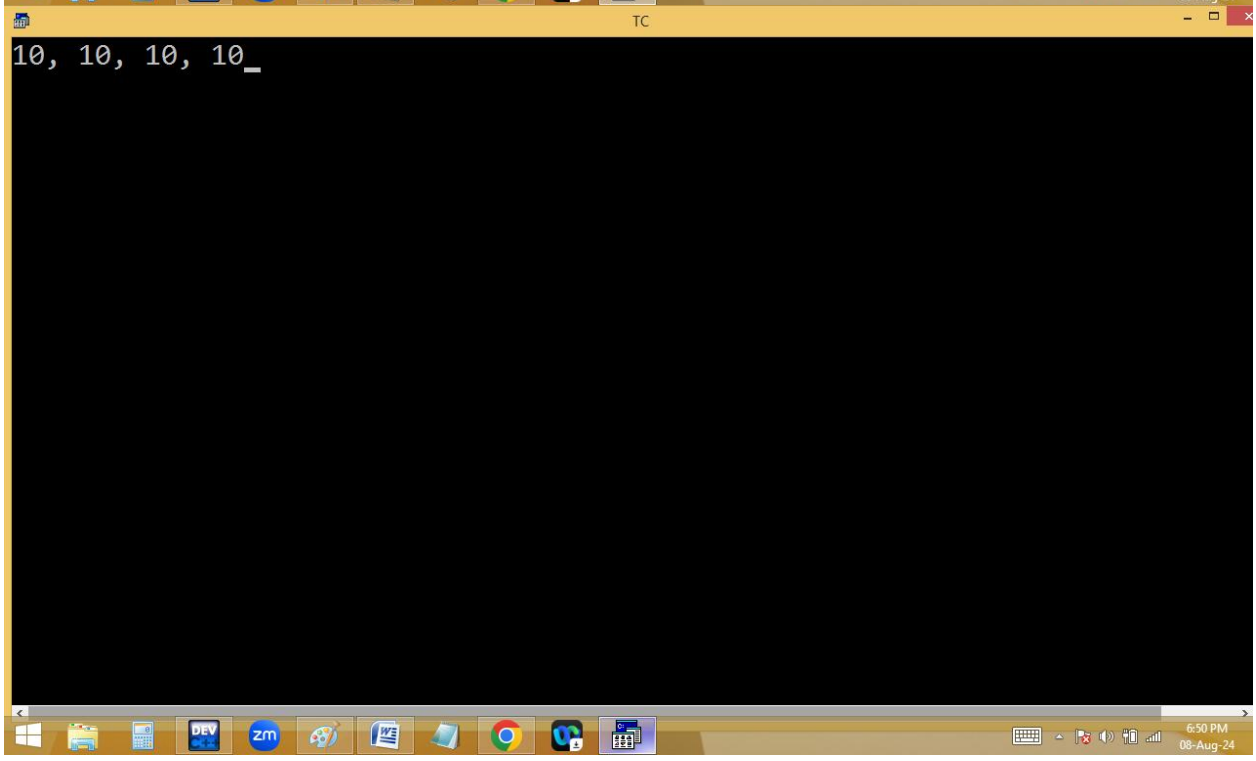

TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 1 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=100; float *p=&a; ;
clrscr();
printf("a=%d",*p);
getch();
}

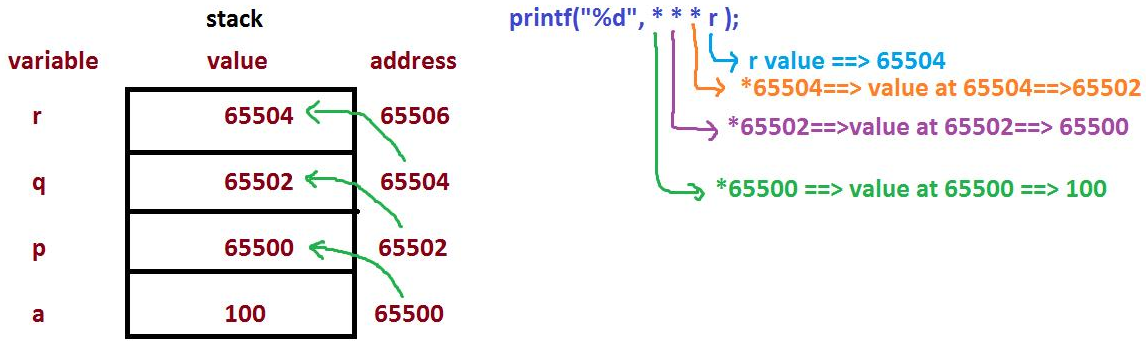

TC
a=0



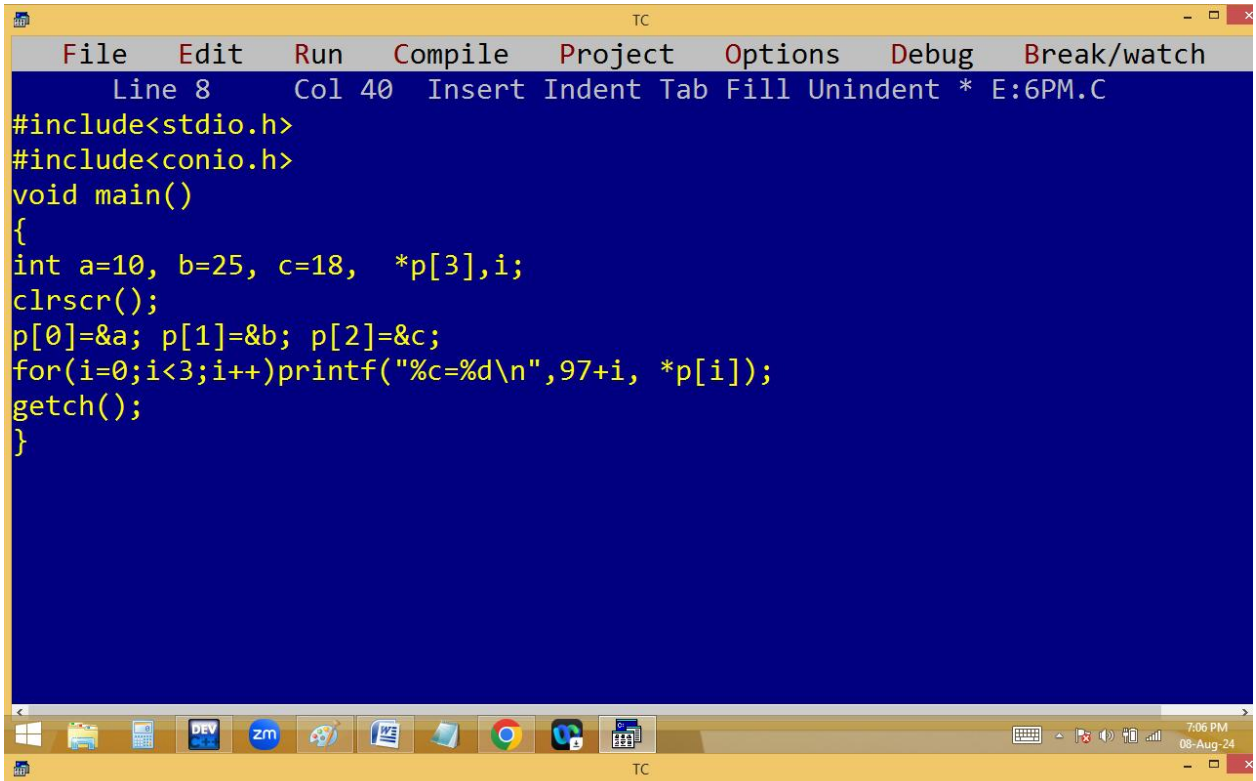
Double pointer / pointer to pointer: The pointer which stores the address of another pointer is called double pointer. They are used to handle dynamic multi dimensional array.

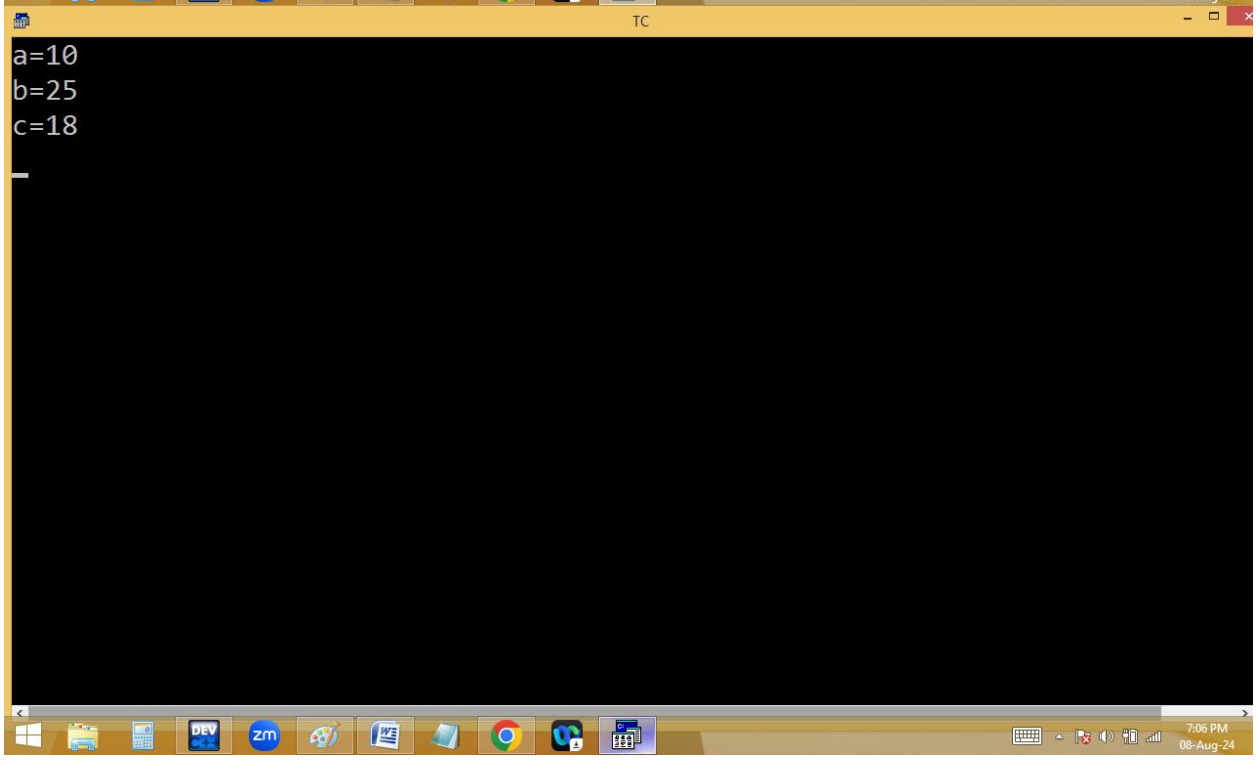

TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 30 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a, **q=&p, ***r=&q ;
clrscr();
printf("%d, %d, %d, %d",a,*p,**q, ***r);
getch();
}


TC
10, 10, 10, 10



Array of pointer:


TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 40 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, b=25, c=18, *p[3],i;
clrscr();
p[0]=&a; p[1]=&b; p[2]=&c;
for(i=0;i<3;i++)printf("%c=%d\n",97+i, *p[i]);
getch();
}


TC
a=10
b=25
c=18

Pointer to array:

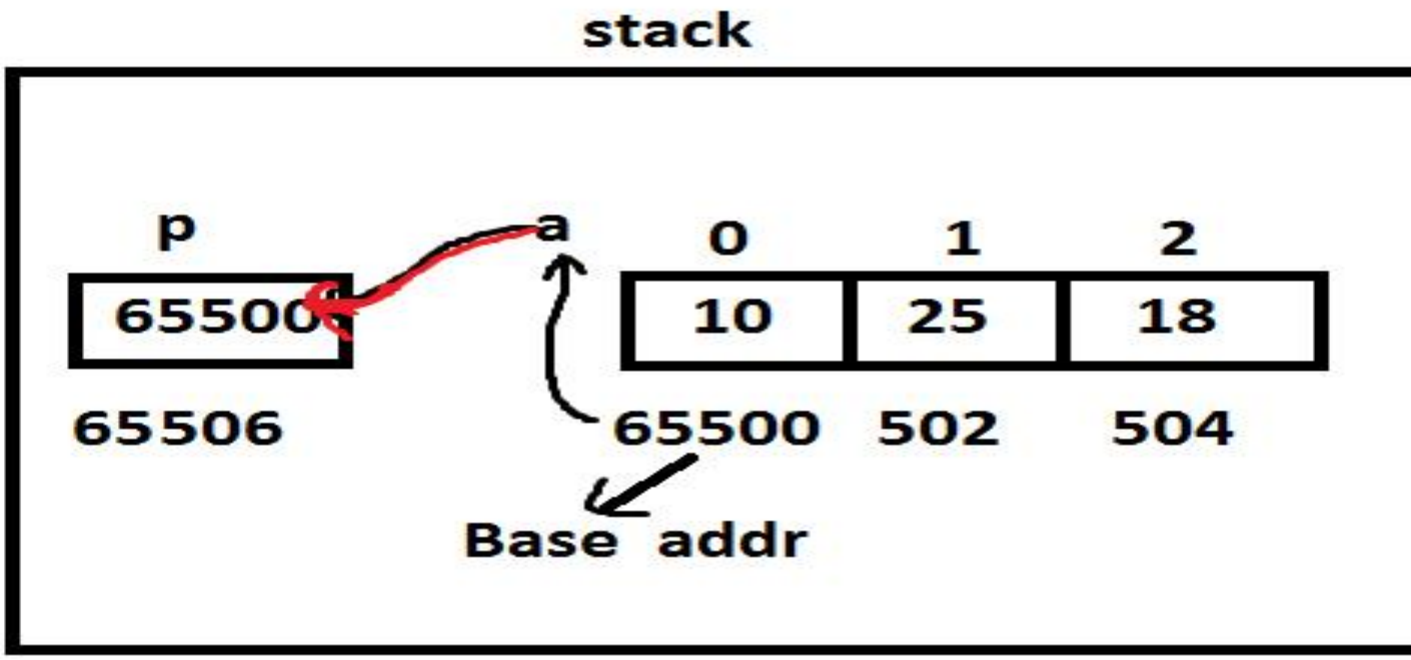
Array is implicit pointer. Due to this it holds the base cell addr [0 cell addr] implicitly. By assigning the array name or 0 cell addr to the pointer, we can handle array elements using the following syntax.

***(ptrvariable + offset/index * sizeof(variable));**

Eg:

int a[3]={10, 25, 18}, *p, i;

p = a ; or p = &a[0]; or p = &a;




```
for(i=0;i<3;i++)
printf("%4d", *(p+i));
```

Here *(p+i) meaning is:

p is 65500

1. *(p+0***2**)➔*65500➔value at 65500➔**10**
2. *(p+1***2**)➔*65502➔value at 65502➔**25**

3. *(p+2*2)➔*65504➔value at 65504➔18

Note: Here 2 is int size.

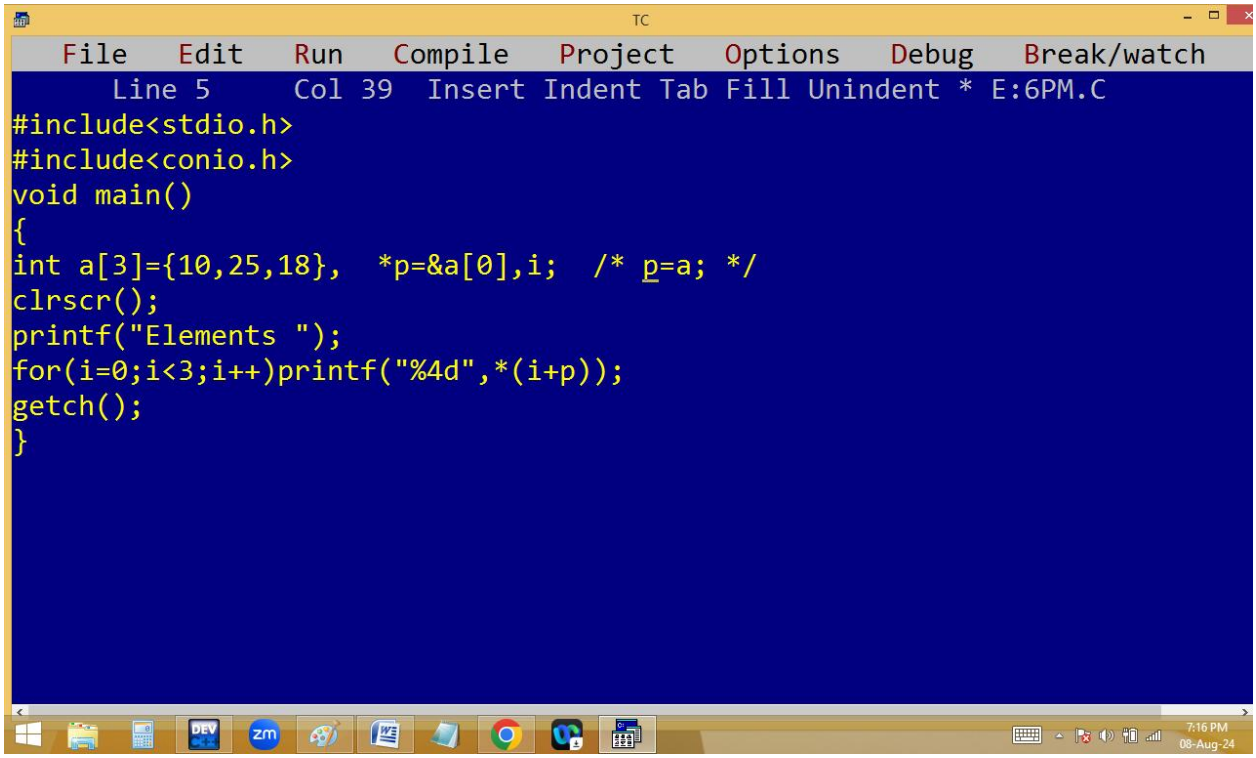
Eg:

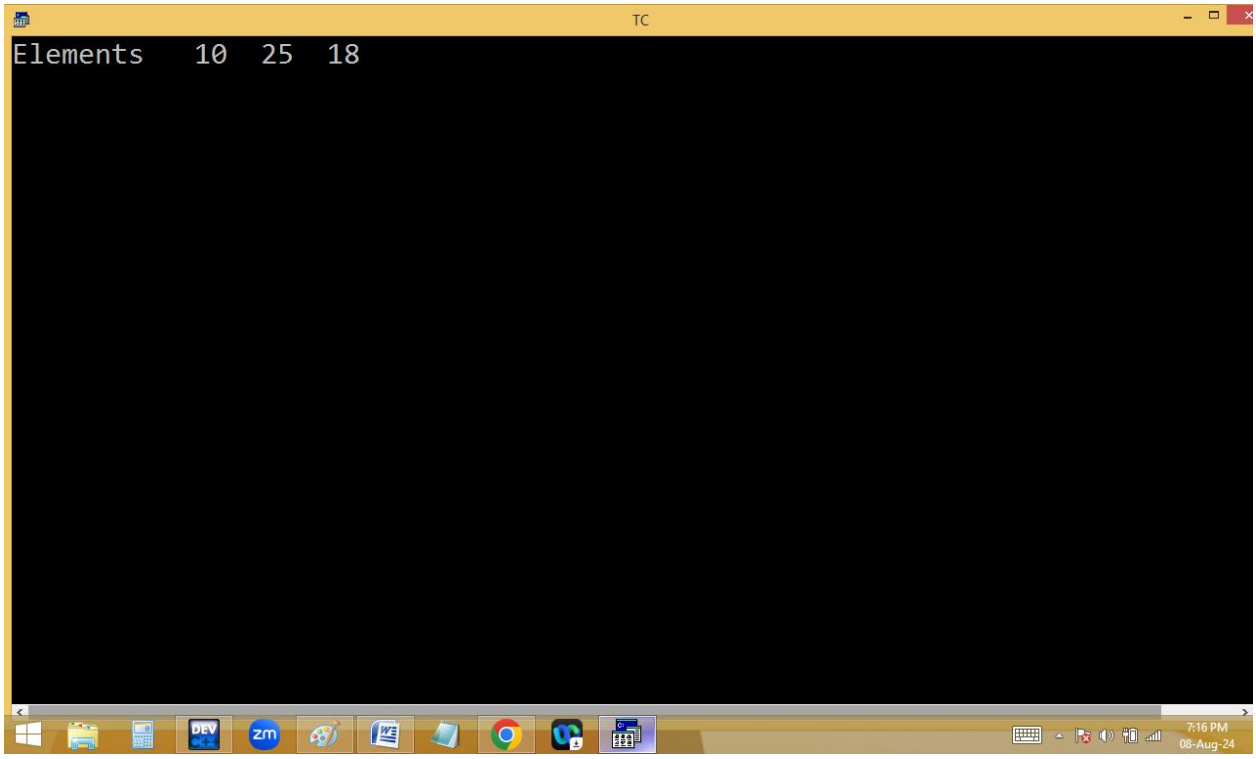
```
#include<stdio.h>
#include<conio.h>
void  main()
{
int  *p,  a[3]={10,25,18}, i;
clrscr();
p = a;   /* p=&a[0]; or p=&a; */
printf("Elements are: ");
for(i=0;i<3;i++)
printf("%4d",*(p+i));
getch();
}
```

Output: Elements are: 10 25 18

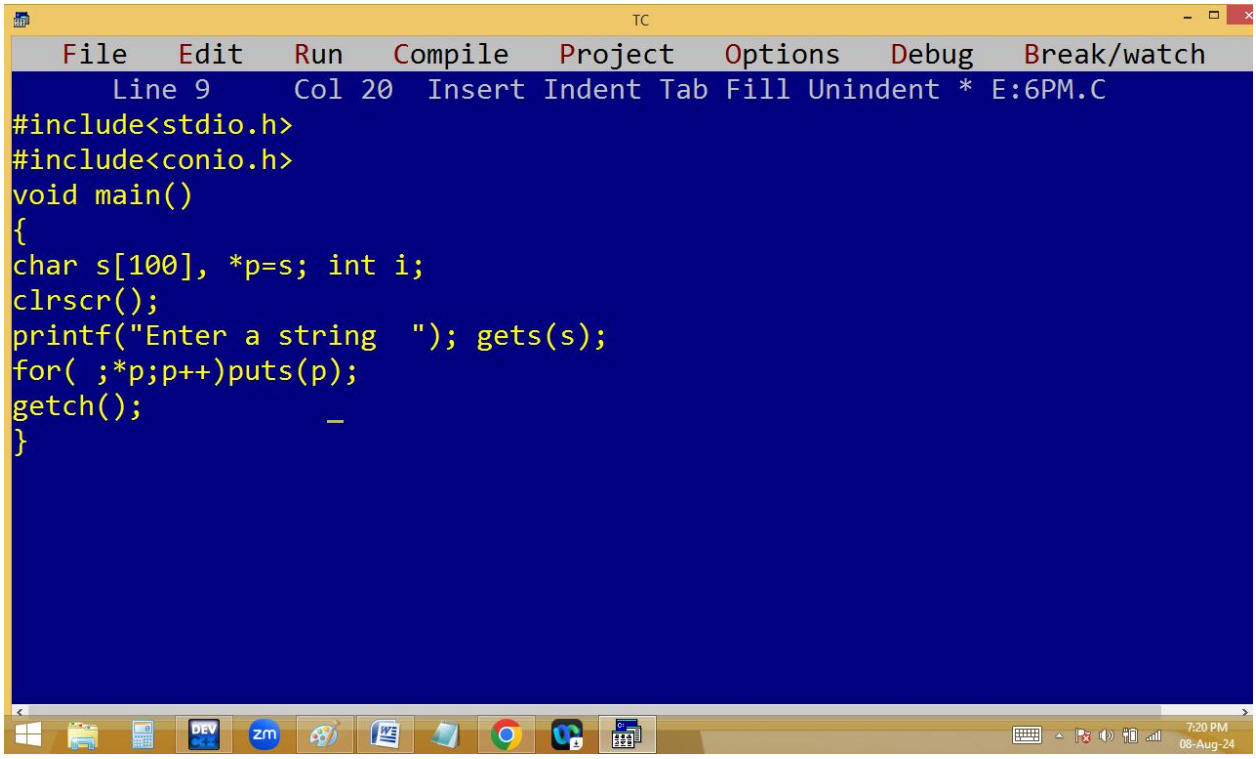
Note: We can access array elements using array / pointer in following ways.

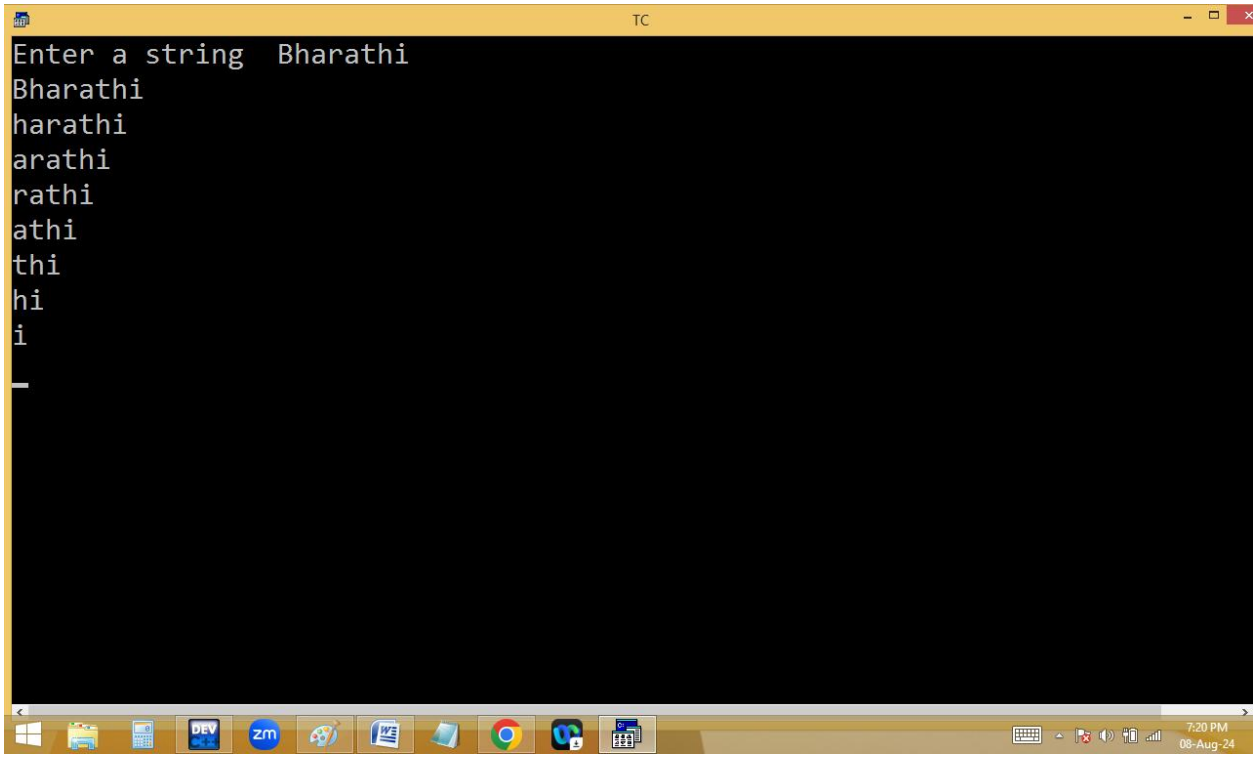
a[i] / i[a] / p[i] / i[p] / *(p+i) / *(a+i) / *(i+p) / *(i+a)

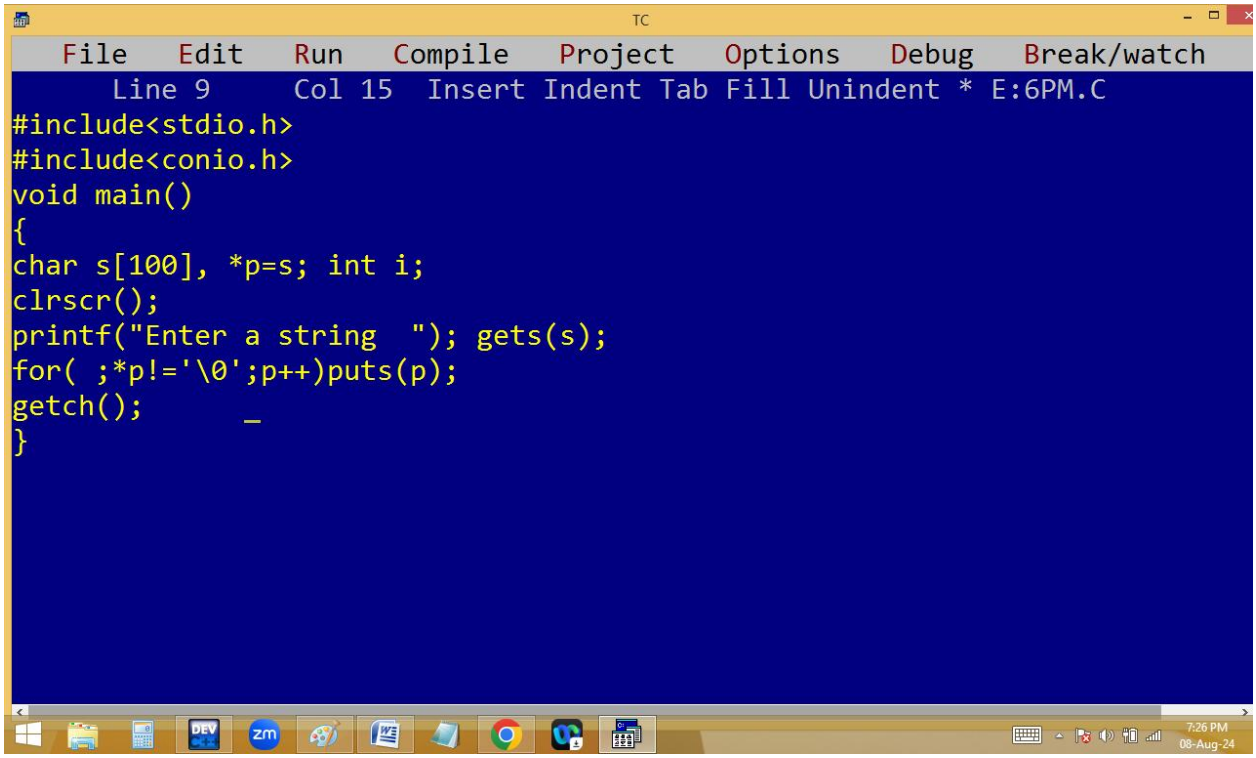

TC
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 39 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3]={10,25,18}, *p=&a[0],i; /* p=a; */
clrscr();
printf("Elements ");
for(i=0;i<3;i++)printf("%4d",*(i+p));
getch();
}

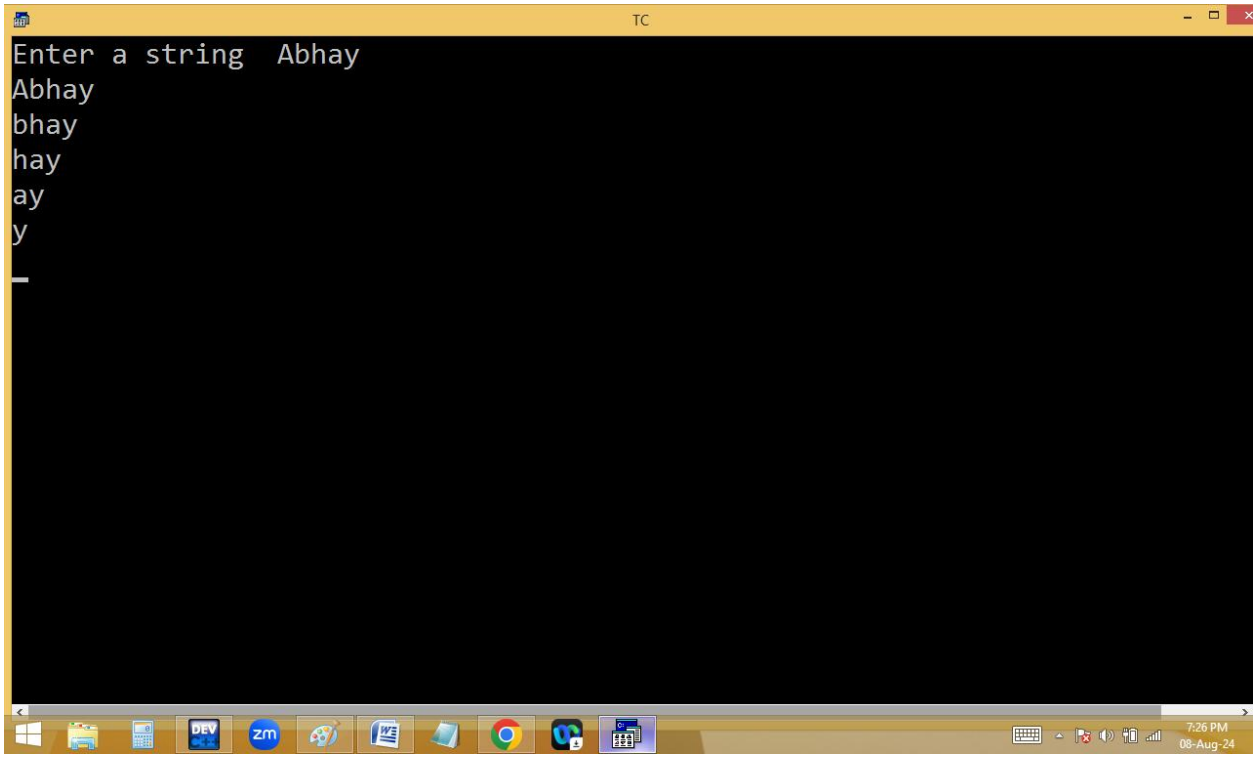

TC
Elements 10 25 18

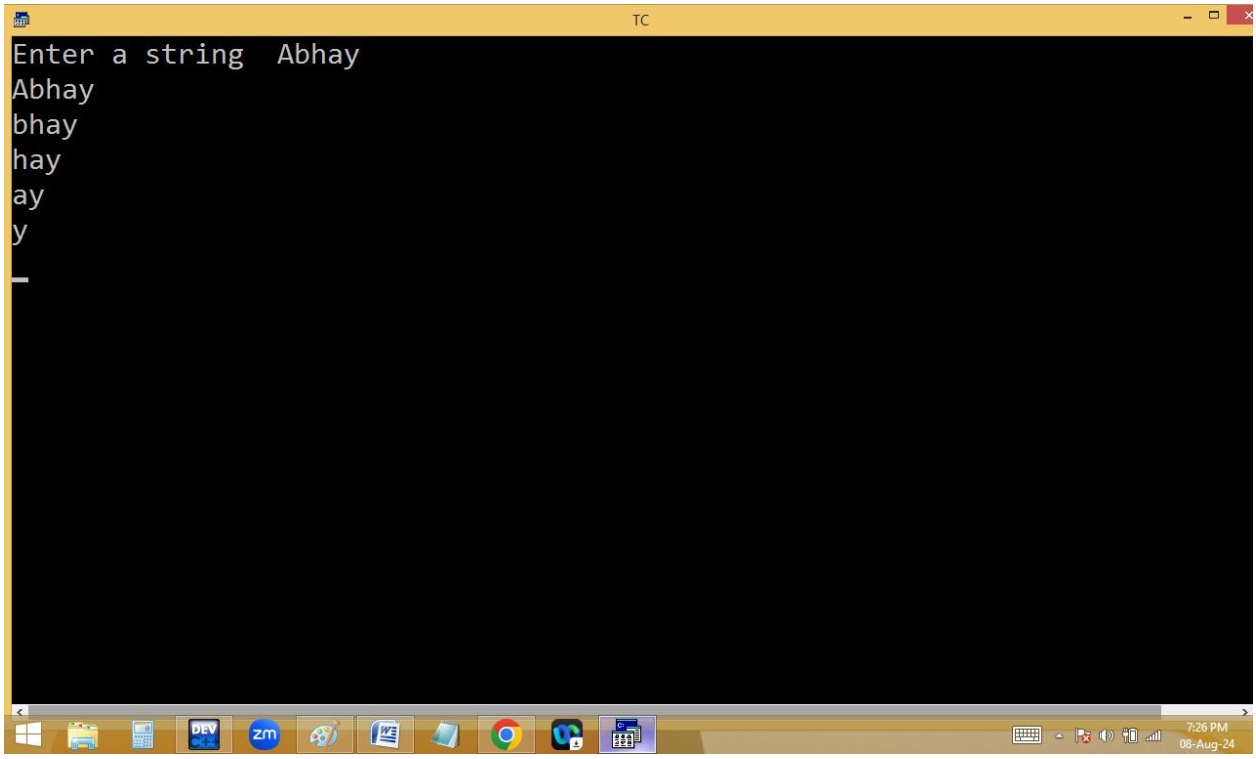
Pointer to string:



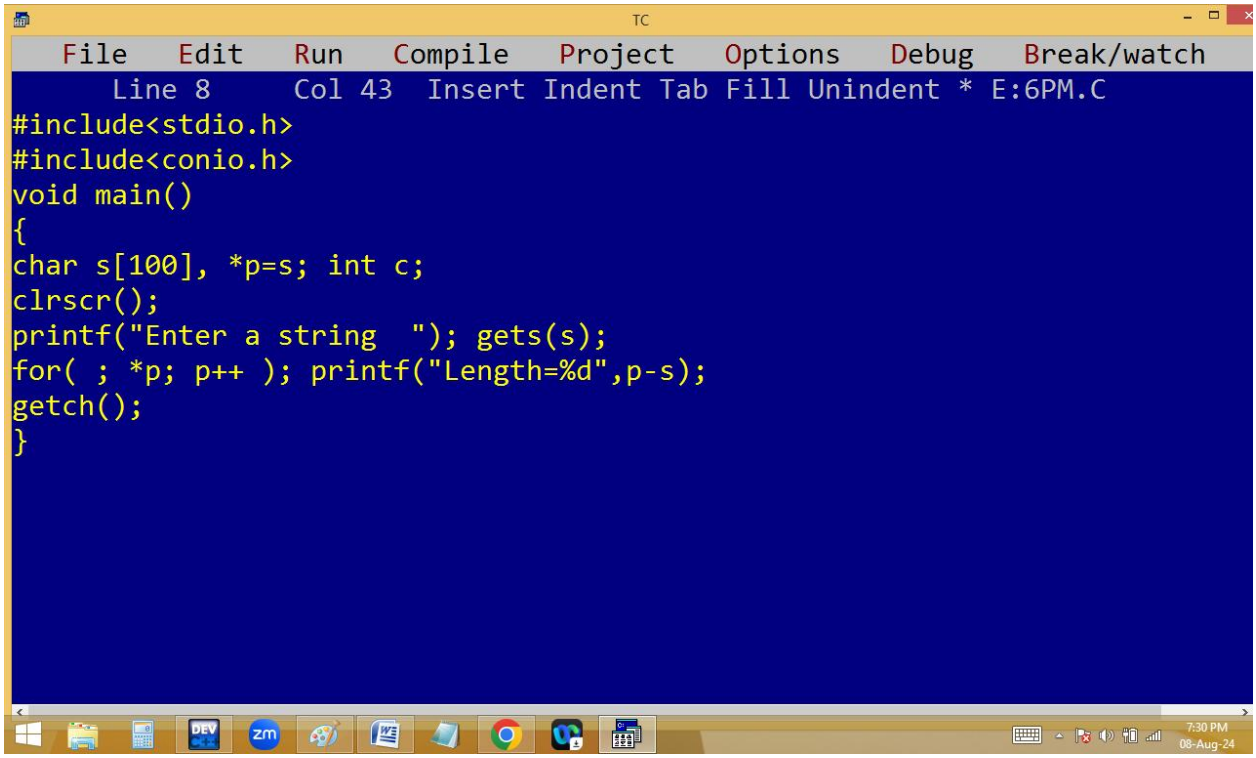


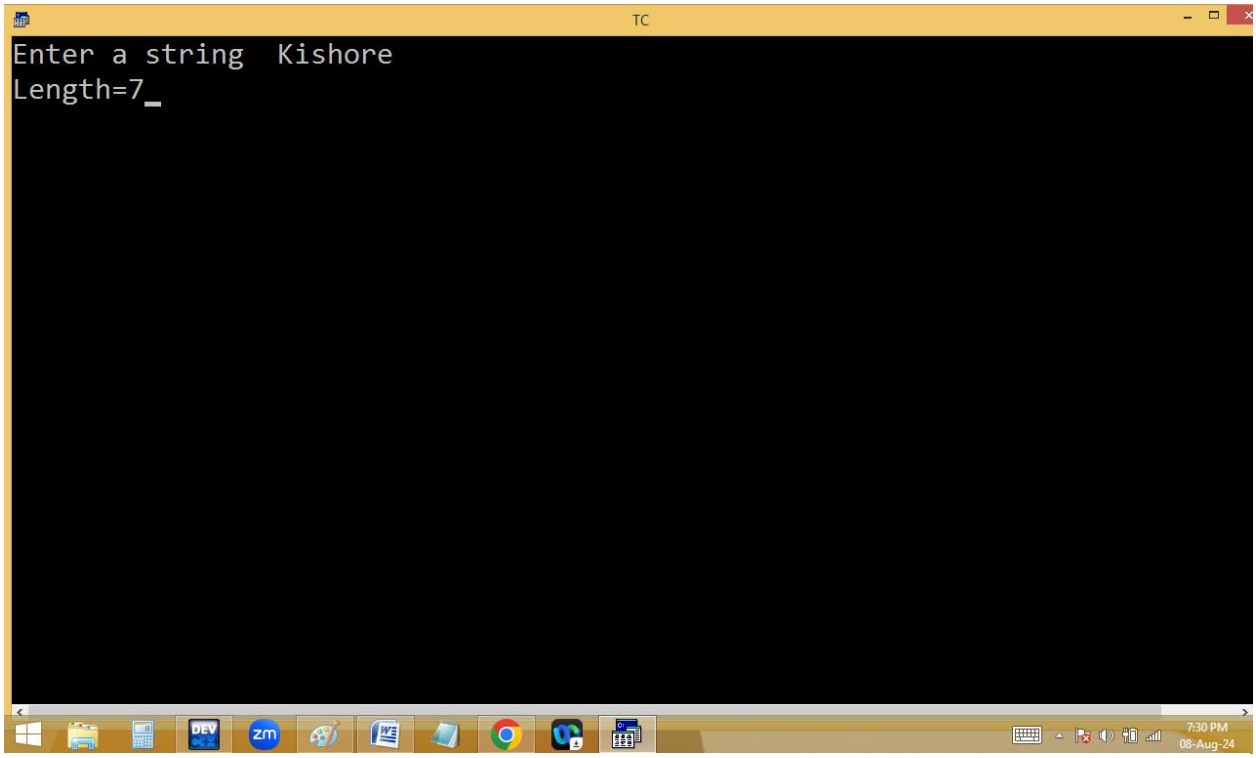

TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 15 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p=s; int i;
clrscr();
printf("Enter a string "); gets(s);
for(;*p!='\0';p++)puts(p);
getch();
}

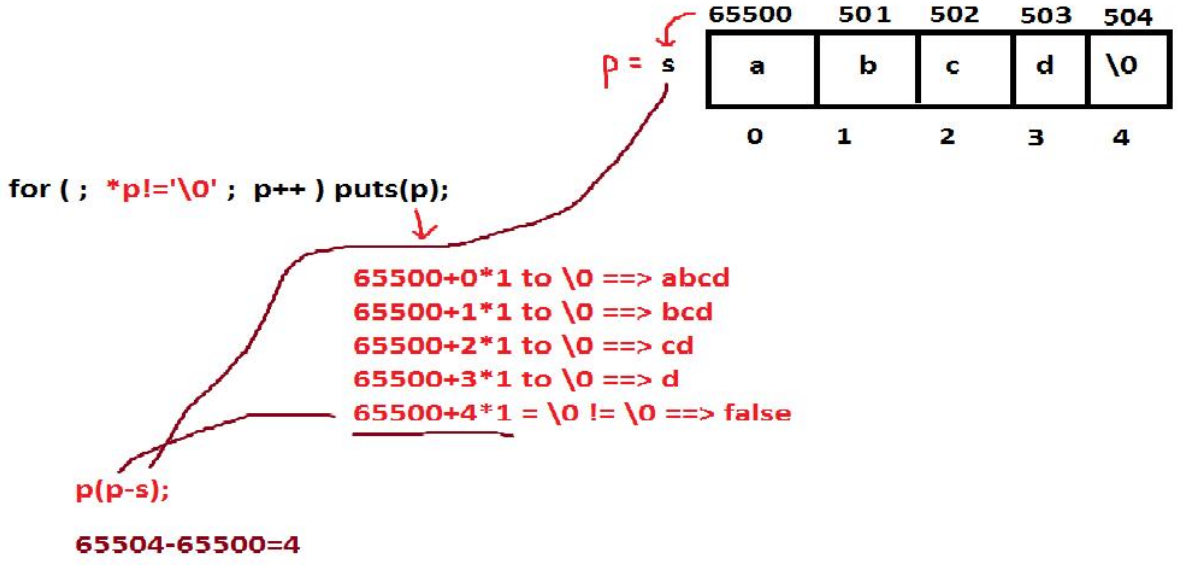

TC
Enter a string Abhay
Abhay
bhay
hay
ay
y



Find the string length using pointers only:


TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 43 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p=s; int c;
clrscr();
printf("Enter a string "); gets(s);
for(; *p; p++); printf("Length=%d",p-s);
getch();
}
7:30 PM
08-Aug-24


TC
Enter a string Kishore
Length=7


65500 501 502 503 504
p = s
a b c d \0
0 1 2 3 4
for (; *p!='\0' ; p++) puts(p);
65500+0*1 to \0 ==> abcd
65500+1*1 to \0 ==> bcd
65500+2*1 to \0 ==> cd
65500+3*1 to \0 ==> d
65500+4*1 = \0 != \0 ==> false
p(p-s);
65504-65500=4