

Pointer to array:

Array is an implicit pointer.

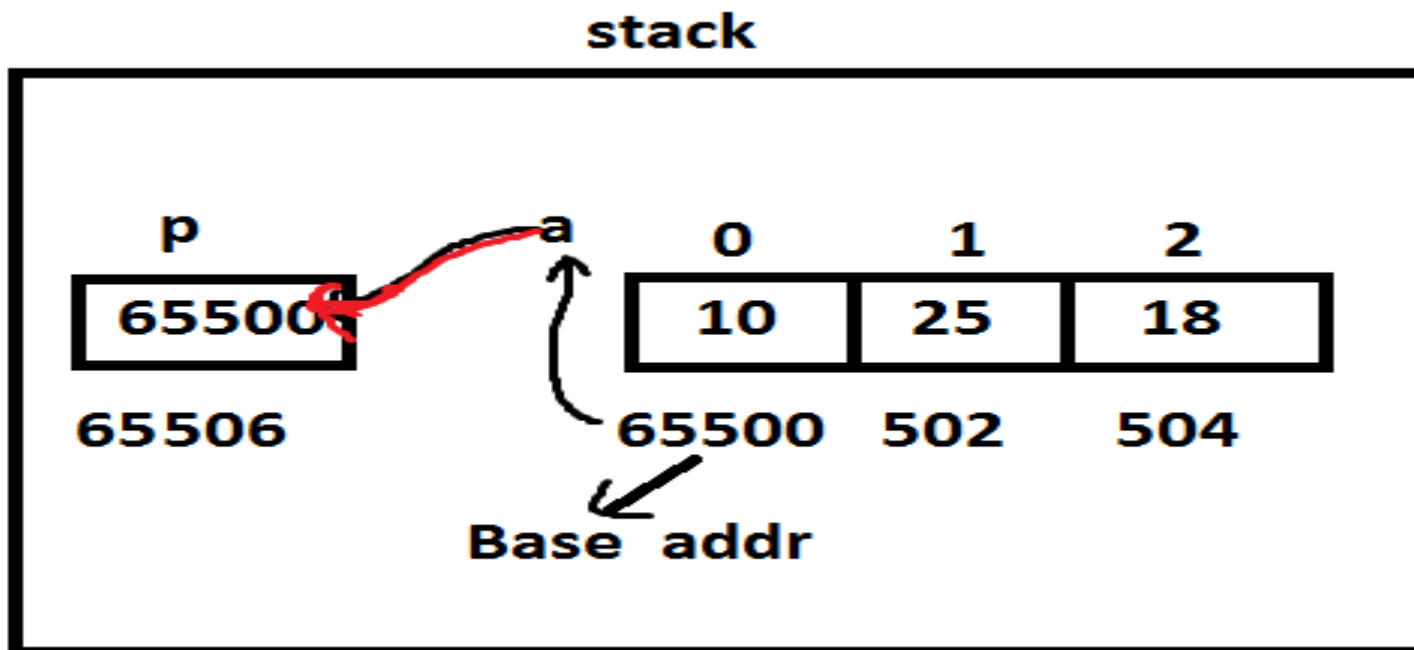
Due to this it holds the base cell addr [0 cell addr] implicitly.

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;
```



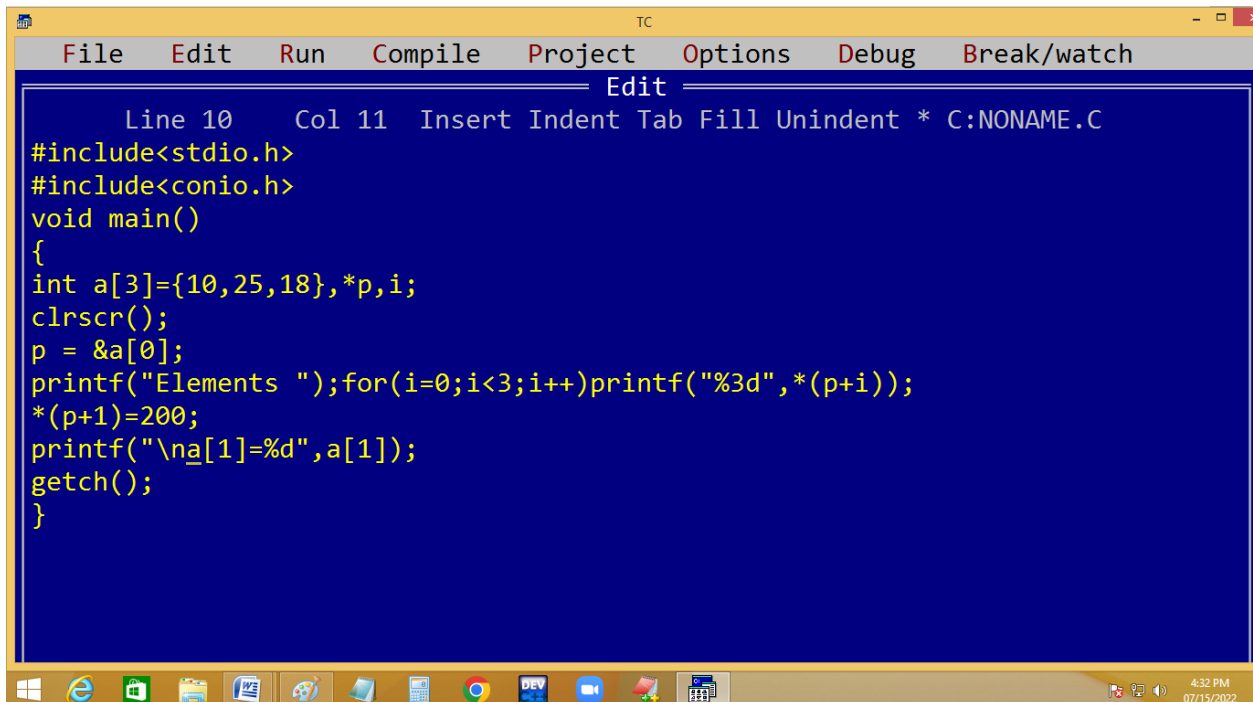
```
for(i=0;i<3;i++)  
printf("%4d", *(p+i));
```

Here $*(p+i)$ meaning is:

p is 65500

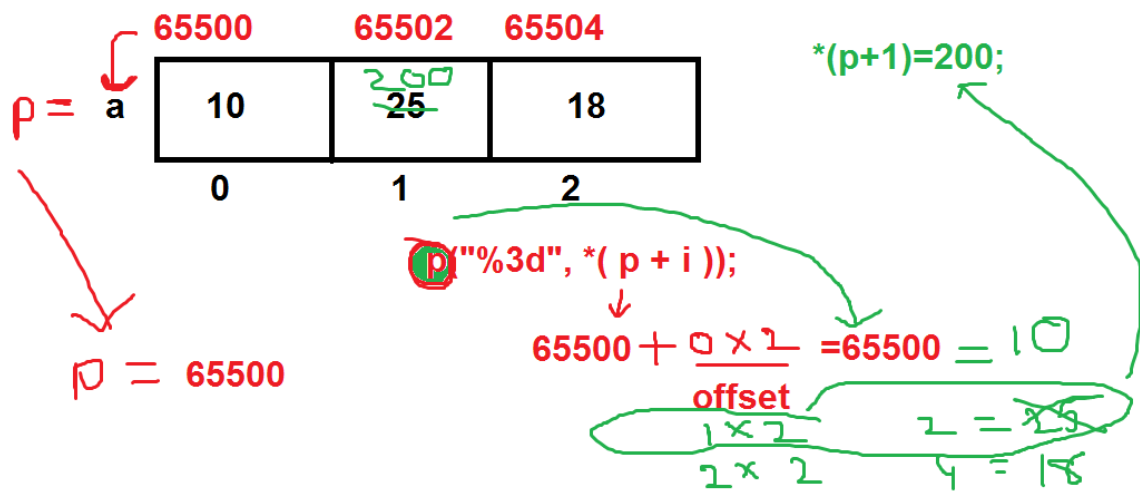
1. $*(p+0*2) \rightarrow *65500 \rightarrow \text{value}$ at
 $65500 \rightarrow 10$
2. $*(p+1*2) \rightarrow *65502 \rightarrow \text{value}$ at
 $65502 \rightarrow 25$
3. $*(p+2*2) \rightarrow *65504 \rightarrow \text{value}$ at
 $65504 \rightarrow 18$

Note: Here 2 is int size.

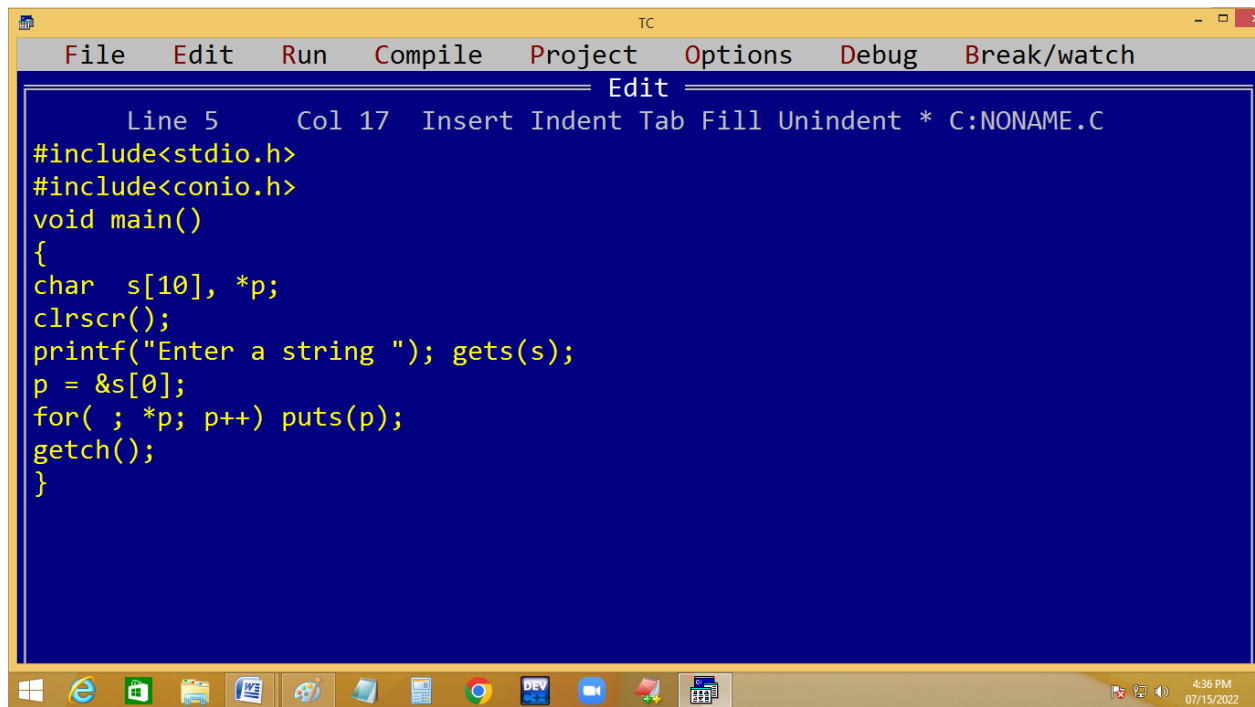


```
TC
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 10 Col 11 Insert Indent Tab Fill Unindent * C:\NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3]={10,25,18},*p,i;
clrscr();
p = &a[0];
printf("Elements ");for(i=0;i<3;i++)printf("%3d",*(p+i));
*(p+1)=200;
printf("\n a[1]=%d",a[1]);
getch();
}
```

```
TC
Elements 10 25 18
a[1]=200
```



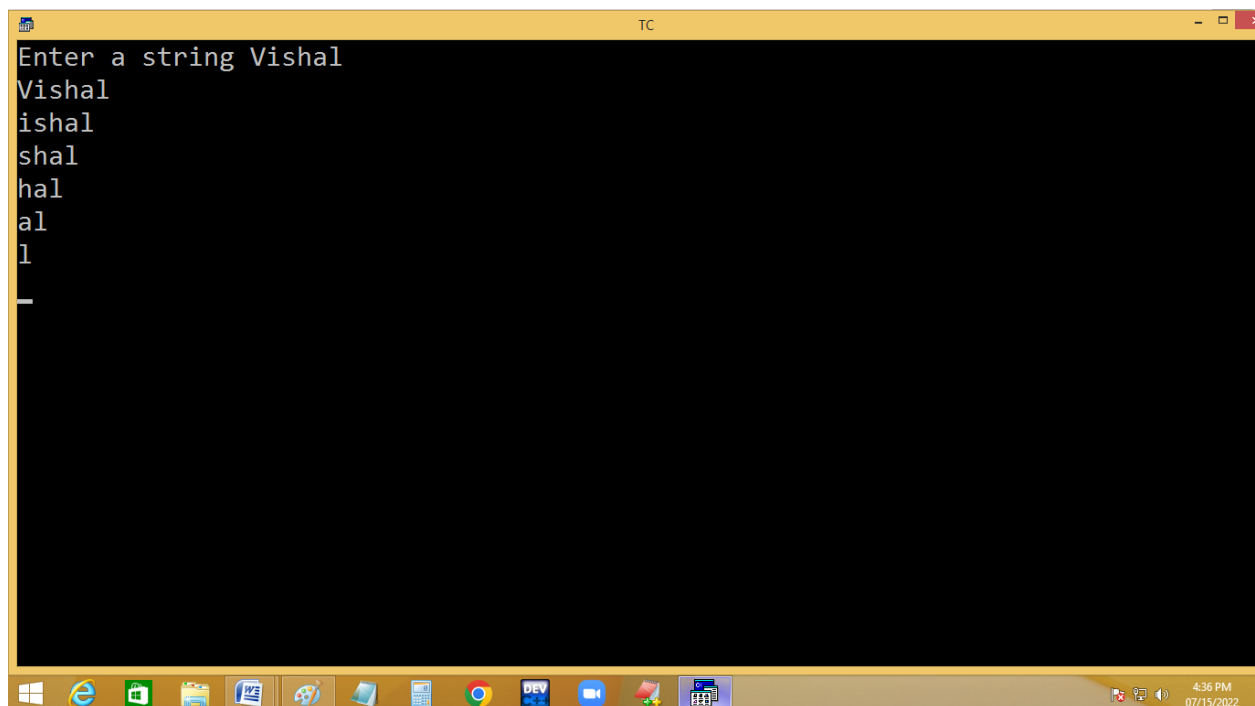
Pointer to string:



The screenshot shows the Turbo C++ IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a toolbar. The main window is titled 'Edit' and contains the following C code:

```
Line 5      Col 17  Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
char  s[10], *p;
clrscr();
printf("Enter a string "); gets(s);
p = &s[0];
for( ; *p; p++) puts(p);
getch();
}
```

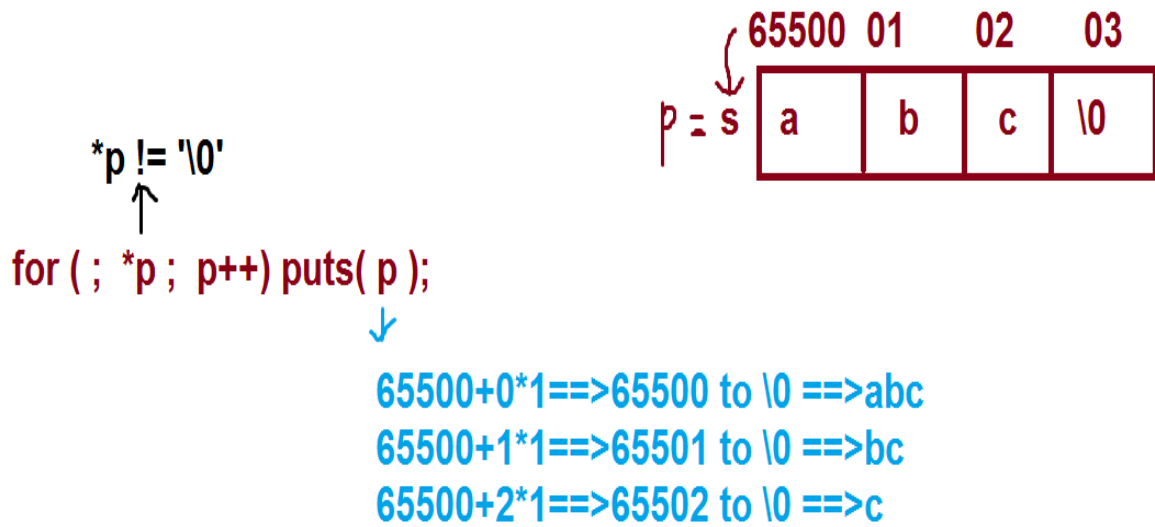
The Windows taskbar at the bottom shows various application icons and the system clock indicating 4:36 PM on 07/15/2022.



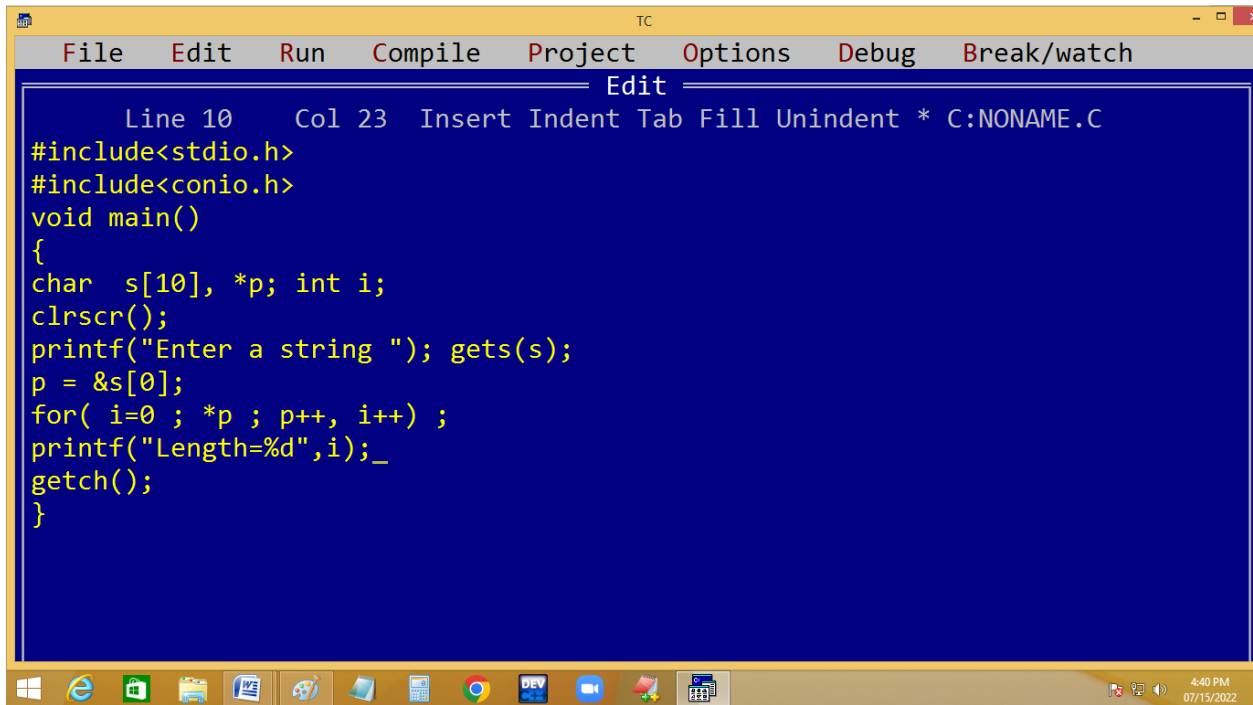
The screenshot shows the Turbo C++ IDE with the same menu bar and toolbar. The main window displays the output of the program:

```
Enter a string Vishal
Vishal
ishal
shal
hal
al
l
_
```

The Windows taskbar at the bottom shows the same application icons and system clock as the first screenshot.



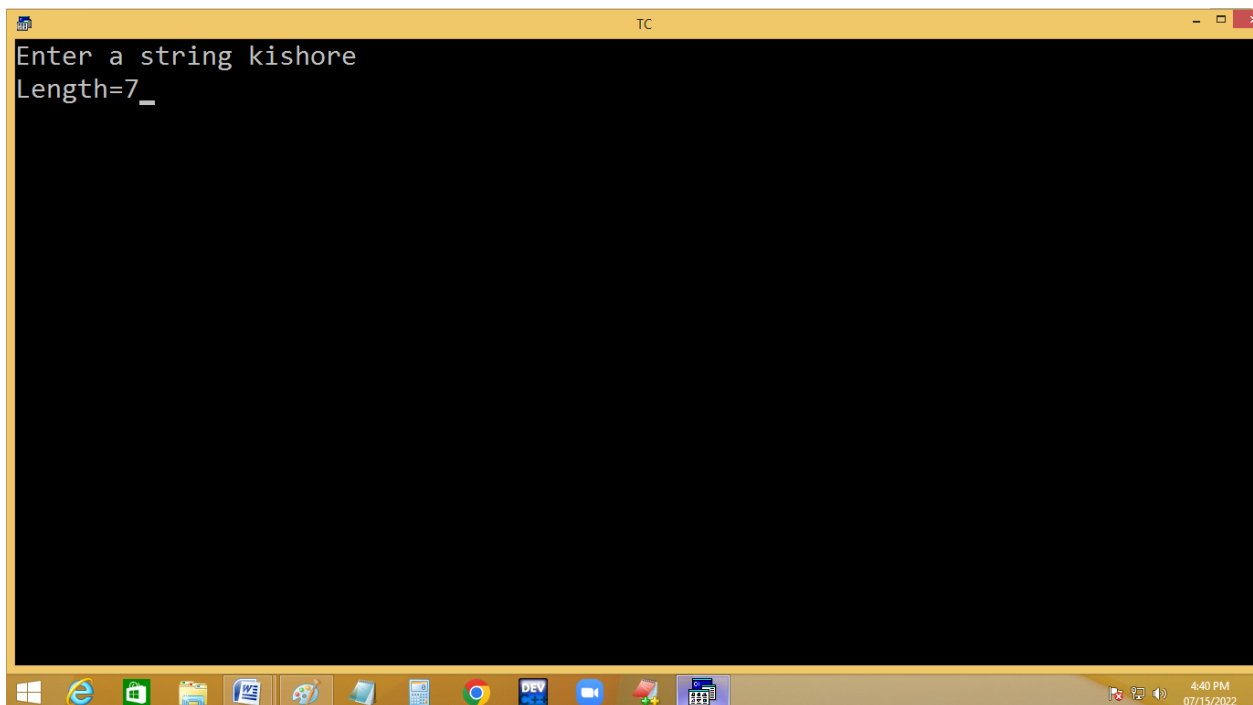
**Eg. finding string length using
pointer.**



The screenshot shows the Turbo C++ IDE with the following code in the editor:

```
Line 10    Col 23  Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
char  s[10], *p; int i;
clrscr();
printf("Enter a string "); gets(s);
p = &s[0];
for( i=0 ; *p ; p++, i++) ;
printf("Length=%d",i);_
getch();
}
```

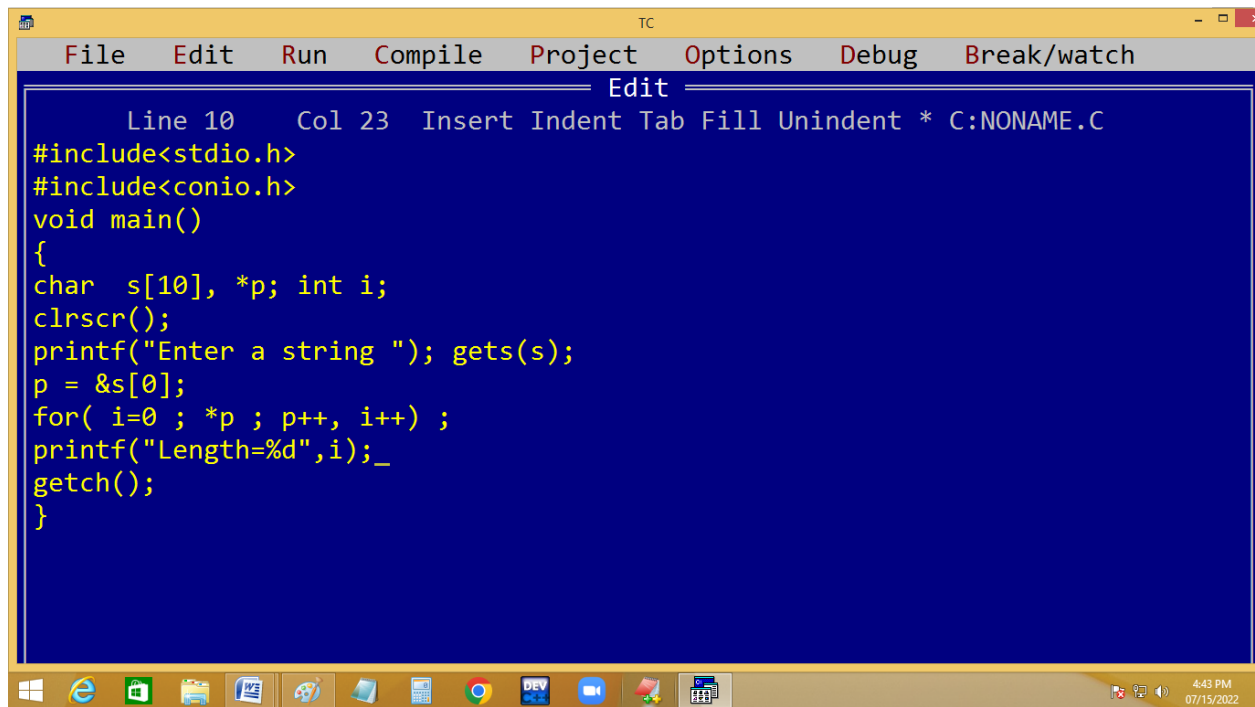
The IDE's menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the bottom right shows the time as 4:40 PM and the date as 07/15/2022.



The screenshot shows the Turbo C++ IDE displaying the output of the program:

```
Enter a string kishore
Length=7_
```

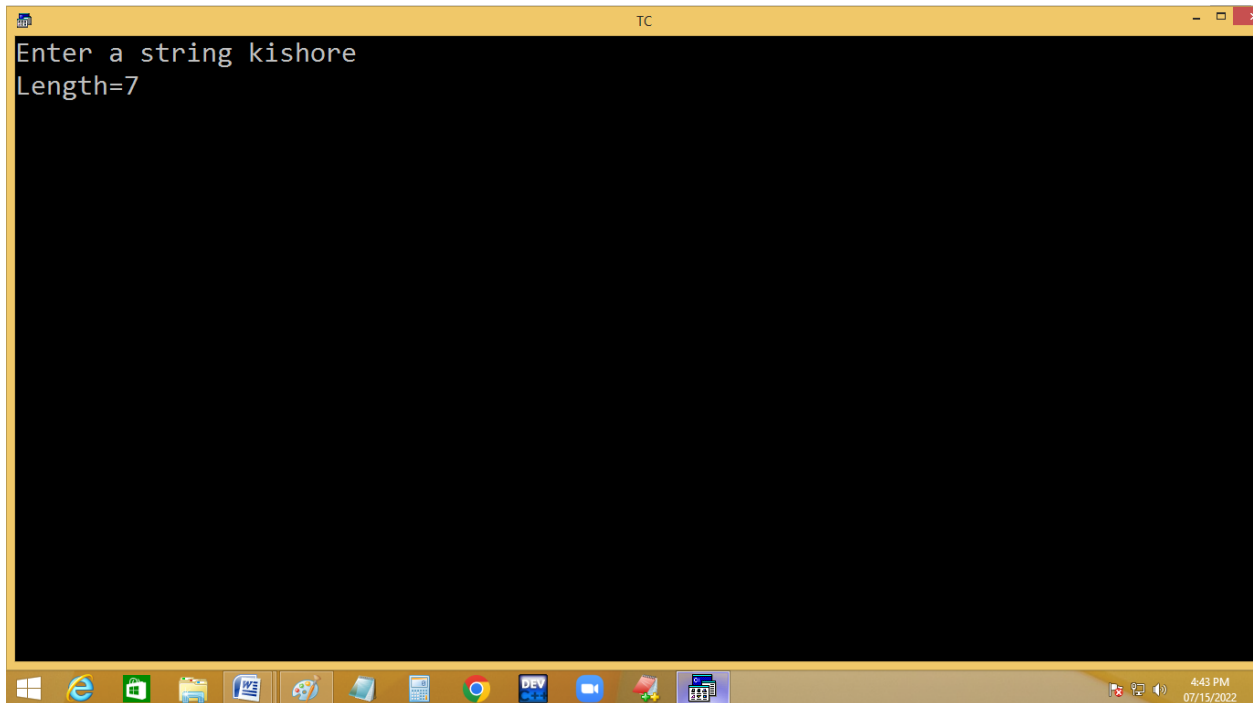
The IDE's status bar at the bottom right shows the time as 4:40 PM and the date as 07/15/2022.



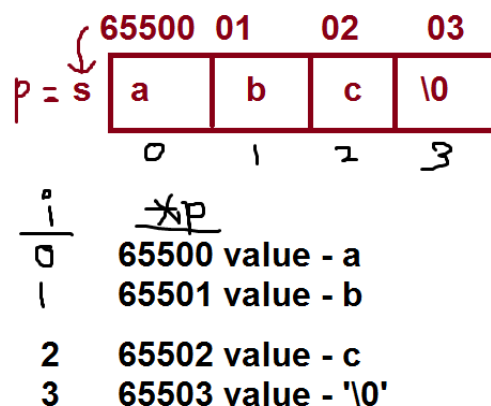
The image shows a screenshot of the Turbo C++ (TC) IDE. The window title is "TC". The menu bar includes "File", "Edit", "Run", "Compile", "Project", "Options", "Debug", and "Break/watch". The "Edit" menu is currently open, showing options: "Line 10", "Col 23", "Insert", "Indent", "Tab", "Fill", "Unindent", and "* C:NONAME.C". The main text area has a dark blue background with yellow text. It contains the following C code:

```
#include<stdio.h>
#include<conio.h>
void main()
{
char  s[10], *p; int i;
clrscr();
printf("Enter a string "); gets(s);
p = &s[0];
for( i=0 ; *p ; p++, i++) ;
printf("Length=%d",i);_
getch();
}
```

The Windows taskbar is visible at the bottom, showing icons for various applications including Internet Explorer, Word, and a calculator. The system clock in the bottom right corner displays "4:43 PM" and "07/15/2022".

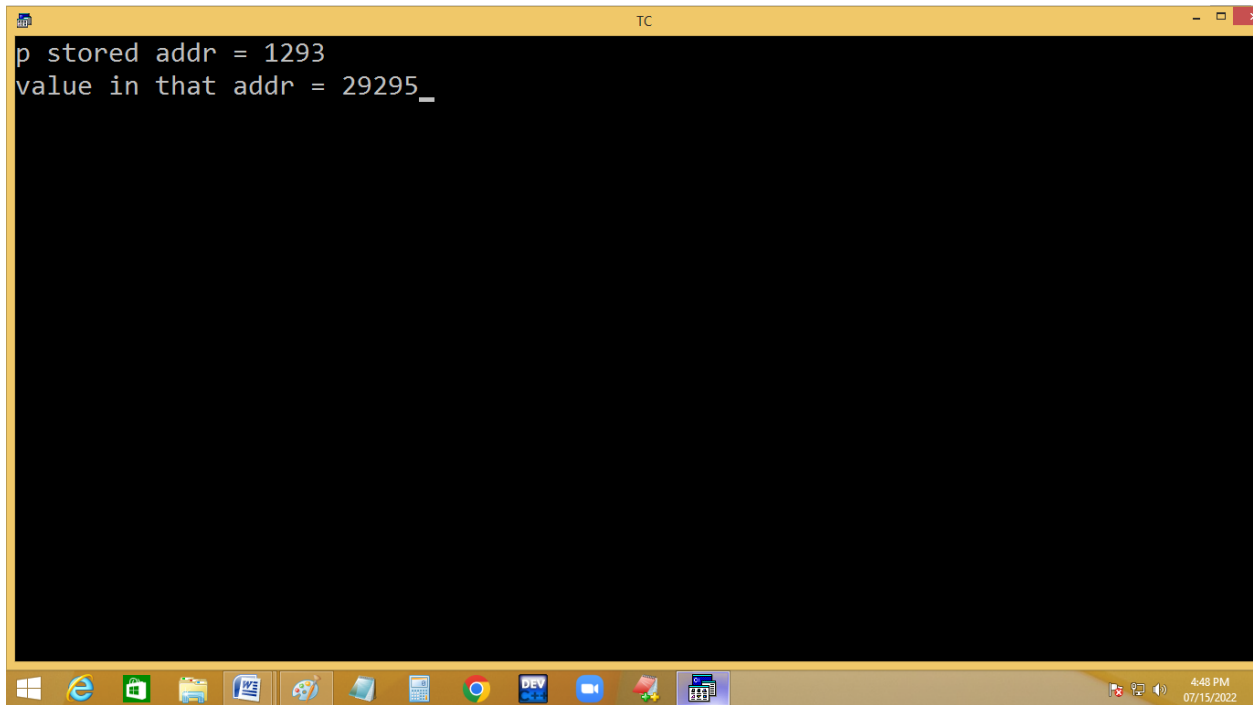


```
for( i=0; *p !='\0'; p++, i++ ) ;  
p( i ); ==> 3
```



Bad / wild pointer:

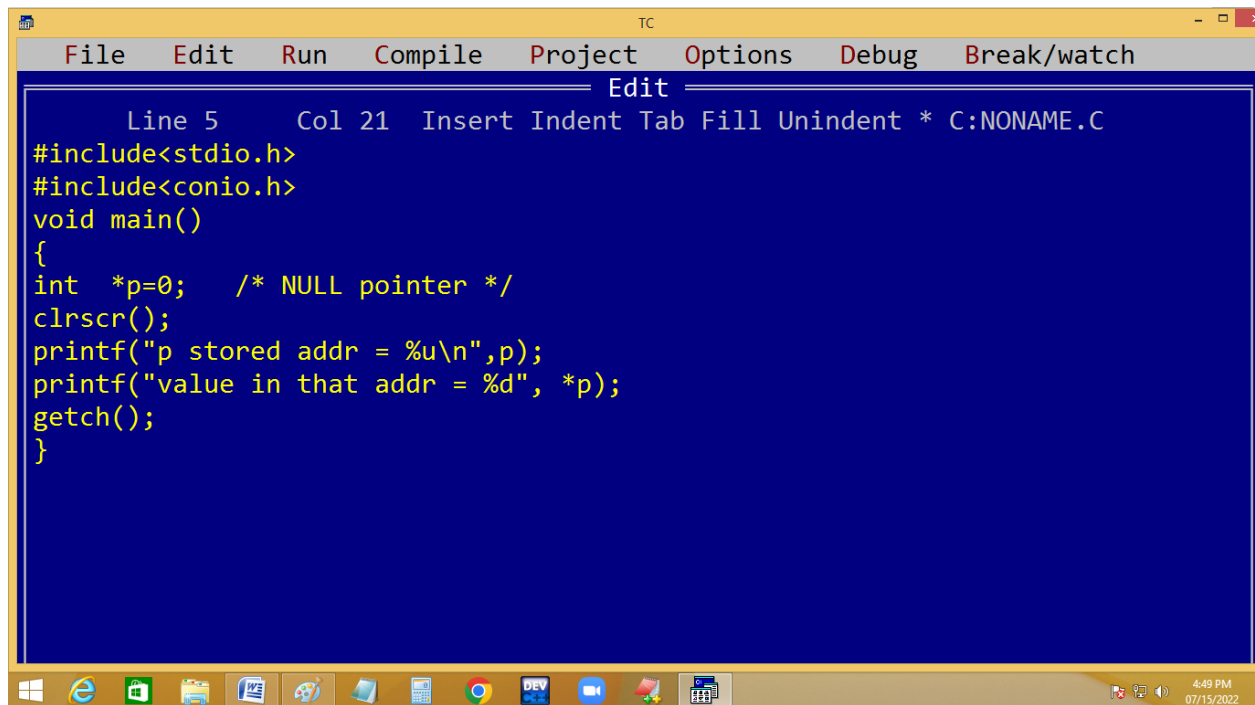
A pointer is declared but not initialized. In this situation the pointer is taking some unknown address and value. This kind of pointer is called bad / wild pointer.



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 two lines of white text: "p stored addr = 1293" and "value in that addr = 29295_". At the bottom of the window is a Windows taskbar with various icons, including the Start button, Internet Explorer, Word, Excel, and several other applications. The system clock in the bottom right corner shows "4:48 PM" and "07/15/2022".

```
p stored addr = 1293
value in that addr = 29295_
```

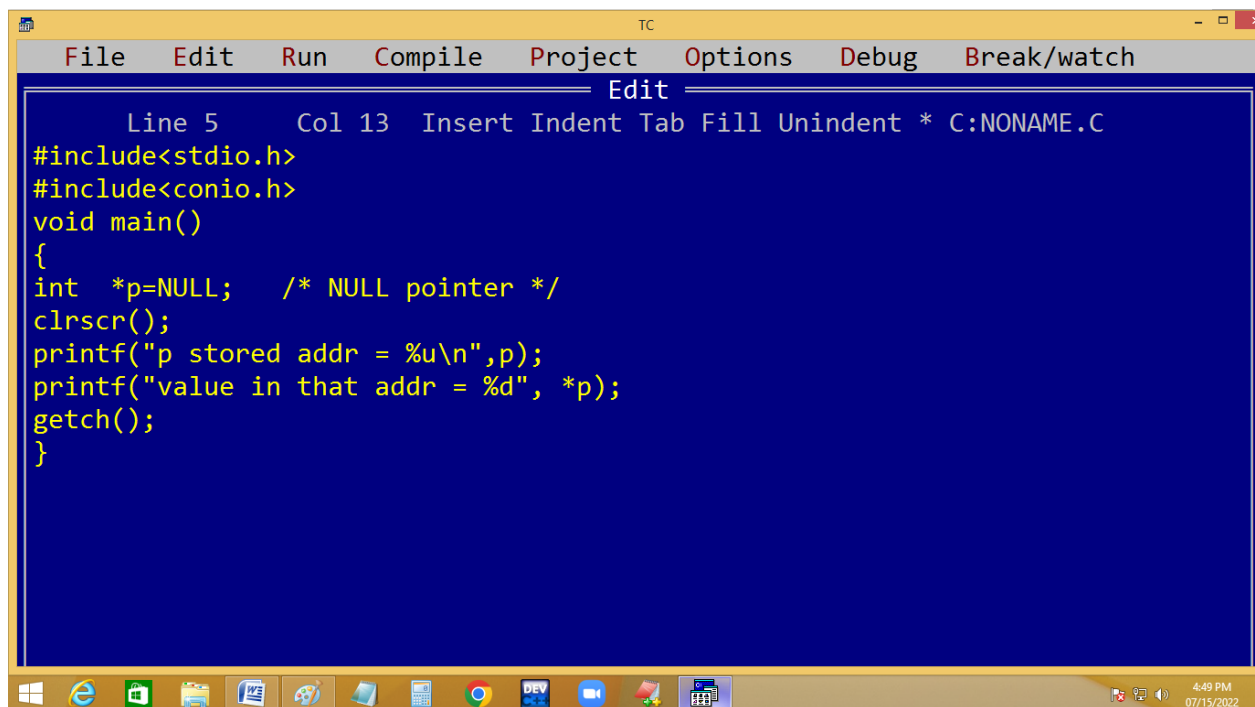
NULL pointer: when a pointer is initialized with 0 or NULL then it is called NULL pointer. To avoid bad and dangling pointers we are using NULL pointer.



The screenshot shows the Turbo C++ IDE with the following code in C:\NONAME.C:

```
Line 5      Col 21  Insert Indent Tab Fill Unindent * C:\NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int  *p=0;   /* NULL pointer */
clrscr();
printf("p stored addr = %u\n",p);
printf("value in that addr = %d", *p);
getch();
}
```

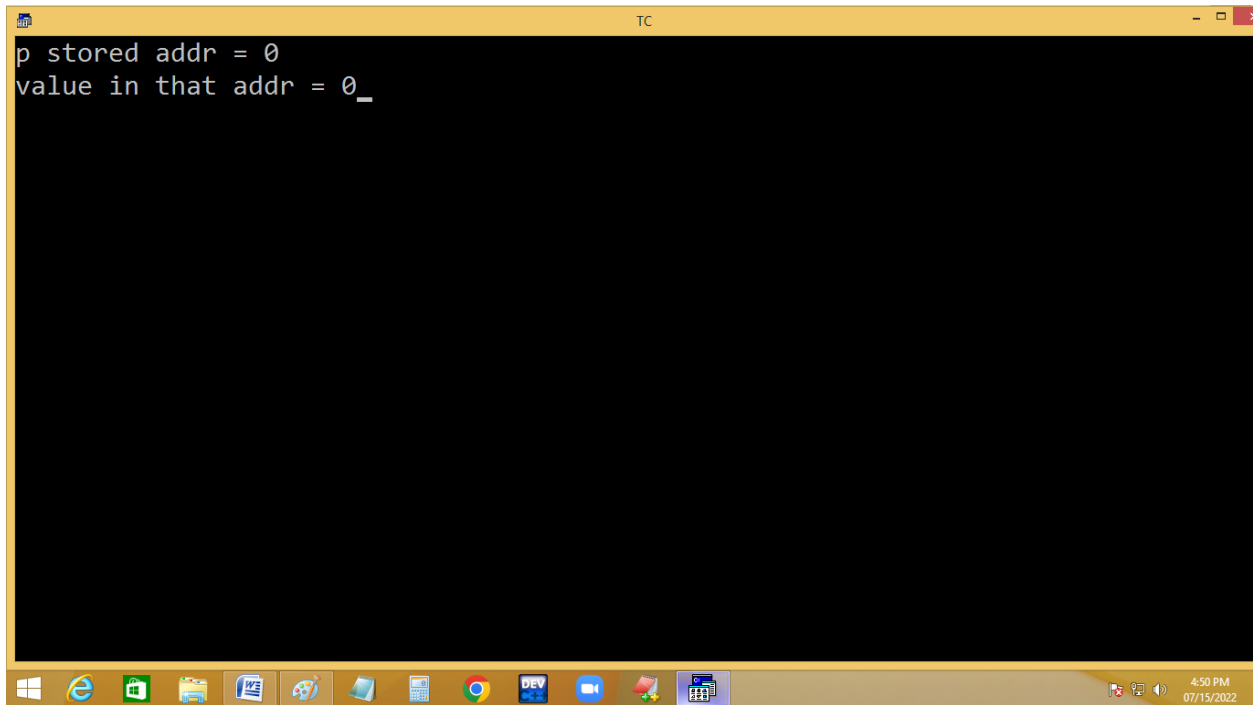
The status bar at the bottom indicates the time is 4:49 PM on 07/15/2022.



The screenshot shows the Turbo C++ IDE with the following code in C:\NONAME.C:

```
Line 5      Col 13  Insert Indent Tab Fill Unindent * C:\NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int  *p=NULL; /* NULL pointer */
clrscr();
printf("p stored addr = %u\n",p);
printf("value in that addr = %d", *p);
getch();
}
```

The status bar at the bottom indicates the time is 4:49 PM on 07/15/2022.

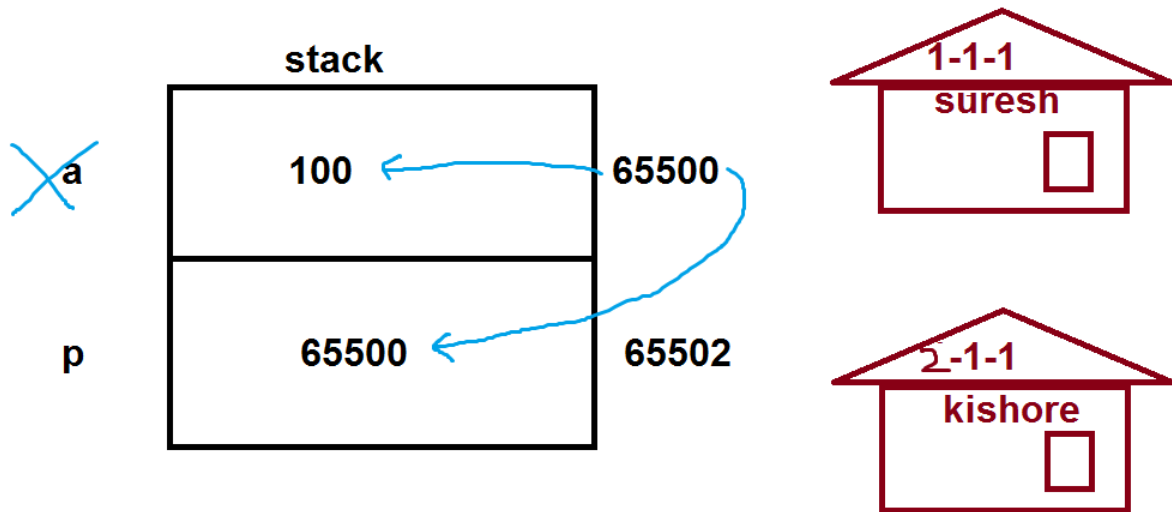


```
p stored addr = 0
value in that addr = 0_
```

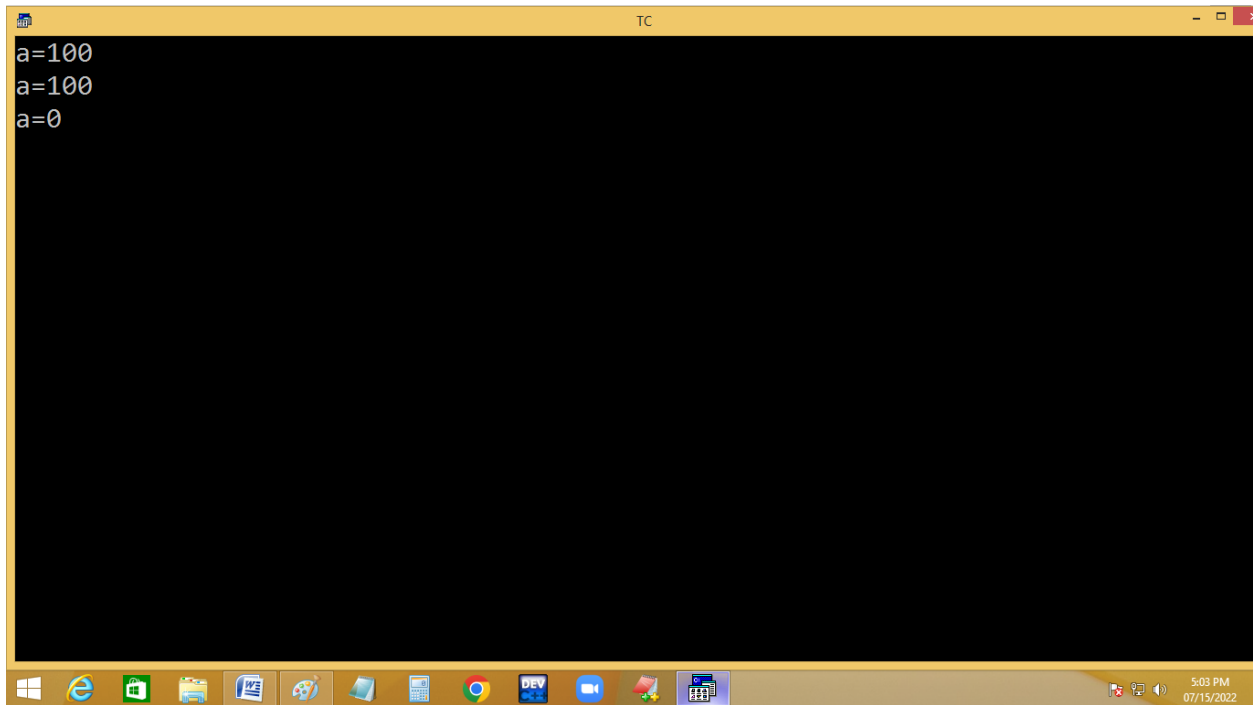
The screenshot shows a Turbo C++ (TC) compiler window with a black background and white text. The code entered is: `p stored addr = 0` followed by `value in that addr = 0_`. The window title bar says 'TC'. The Windows taskbar is visible at the bottom with various icons and a system clock showing 4:50 PM on 07/15/2022.

Dangling pointer:

A pointer is declared and some variable / memory address is assigned. After some time that variable or memory released [deleted]. But still the pointer is storing the deleted variable or memory address. This kind of pointer is called dangling pointer. To avoid this we are using NULL pointer.



```
TC
File Edit Run Compile Project Options Debug Break/watch
Edit
Line 1 Col 19 Insert Indent Tab Fill Unindent * C:NONAME.C
#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);
p=NULL;
printf("a=%d",*p);
getch();
}
```



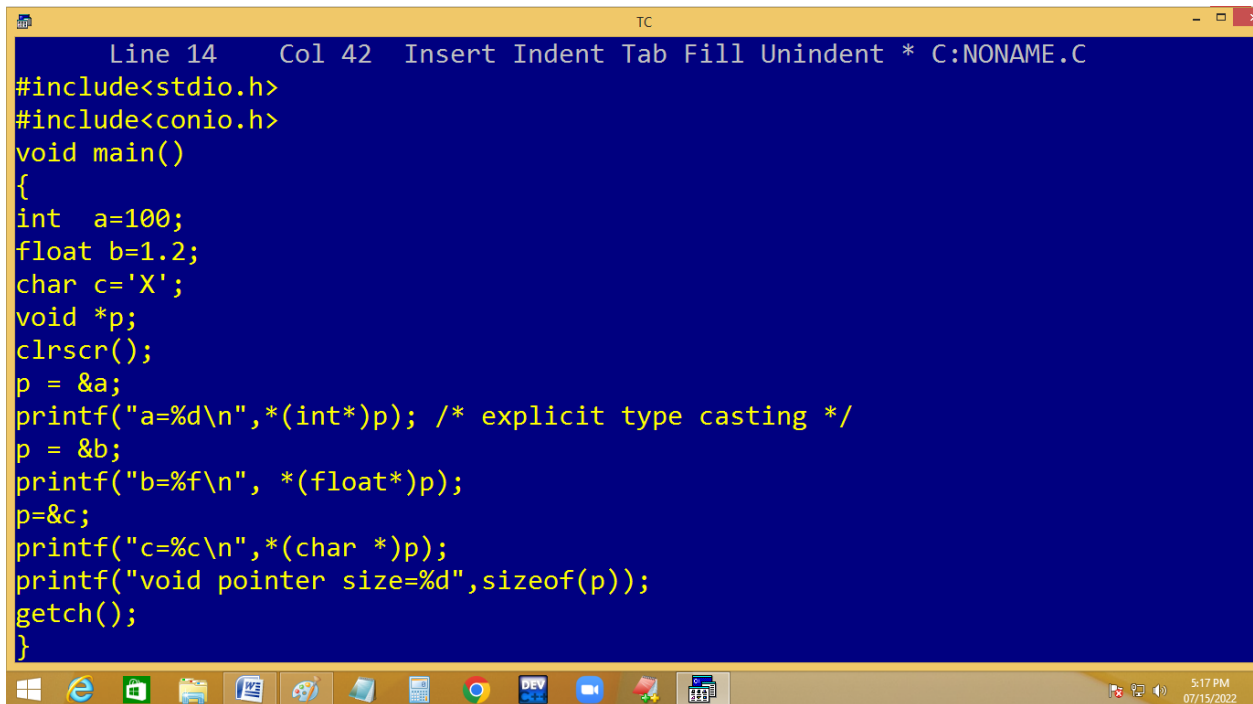
```
a=100
a=100
a=0
```

void/generic pointer:

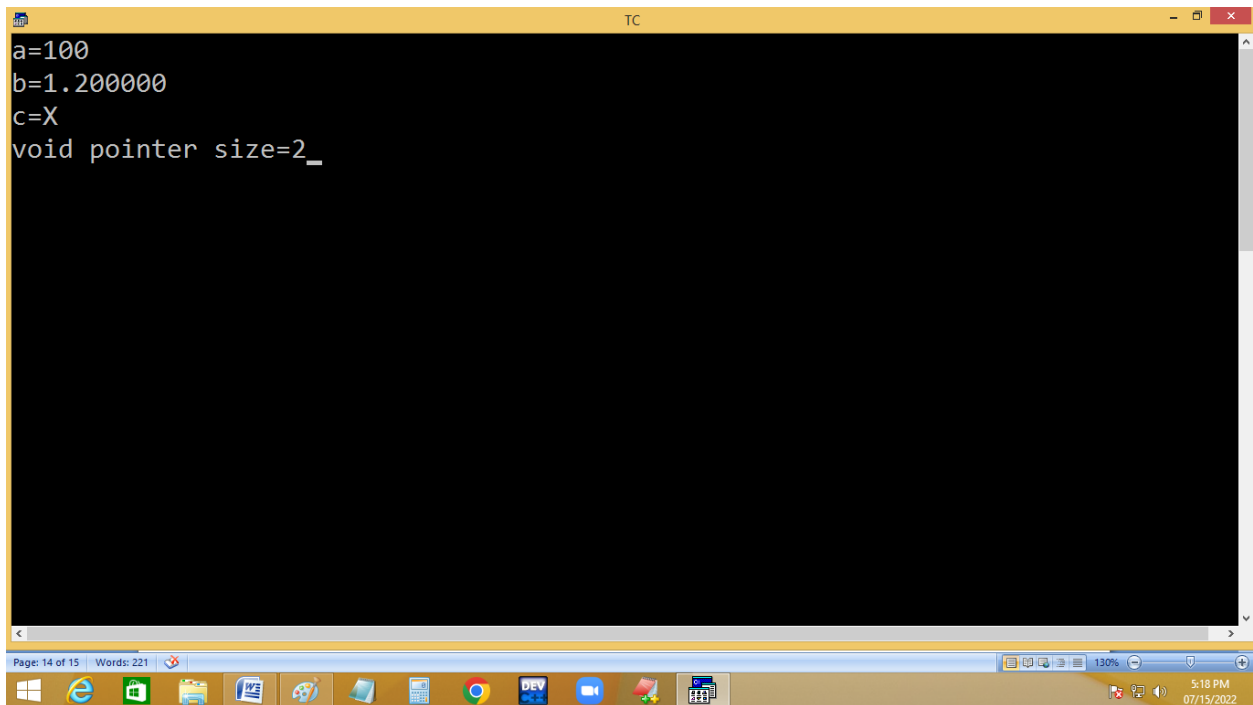
pointer is used to store the address of another variable / memory. When several variables with different type, we have to declare several pointers.

void pointer stores any type of address. But before going to access void pointer, explicit type casting should be provided.

Void pointer takes 2 bytes. It is very much used in dynamic memory allocation.



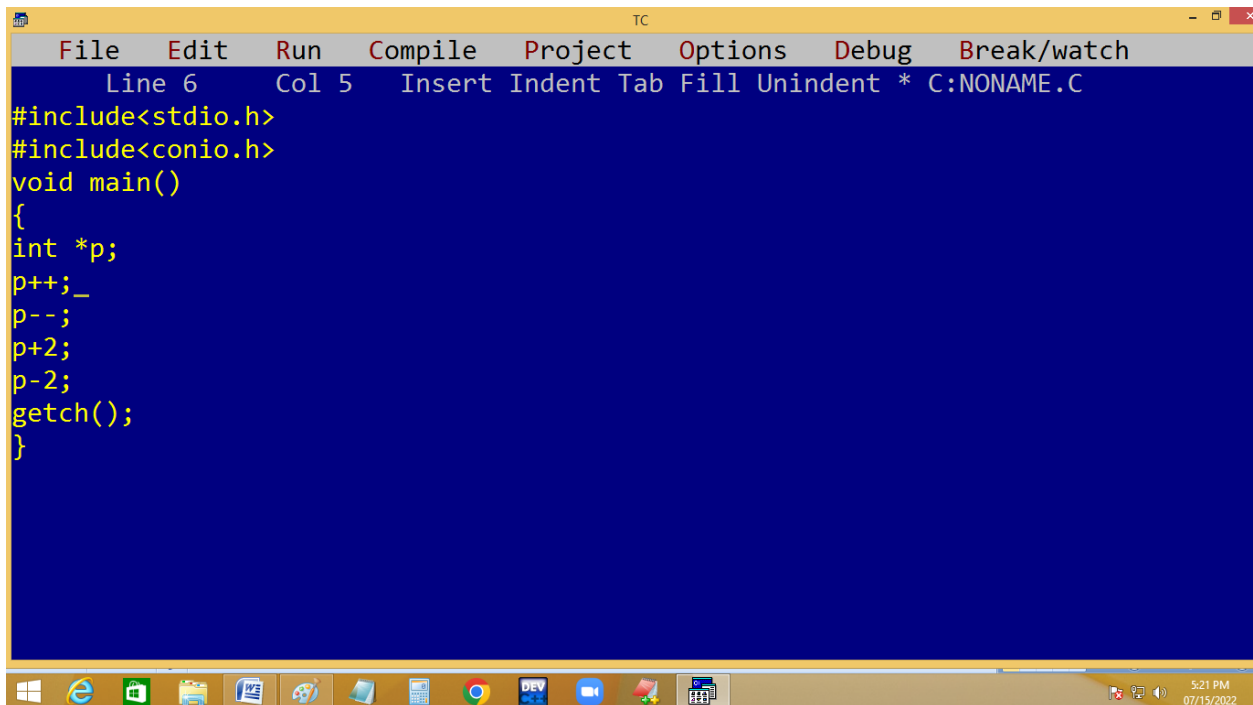
```
Line 14 Col 42 Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a=100;
float b=1.2;
char c='X';
void *p;
clrscr();
p = &a;
printf("a=%d\n",*(int*)p); /* explicit type casting */
p = &b;
printf("b=%f\n", *(float*)p);
p=&c;
printf("c=%c\n",*(char *)p);
printf("void pointer size=%d",sizeof(p));
getch();
}
```



```
a=100
b=1.200000
c=X
void pointer size=2_
```

Pointer arithmetic:

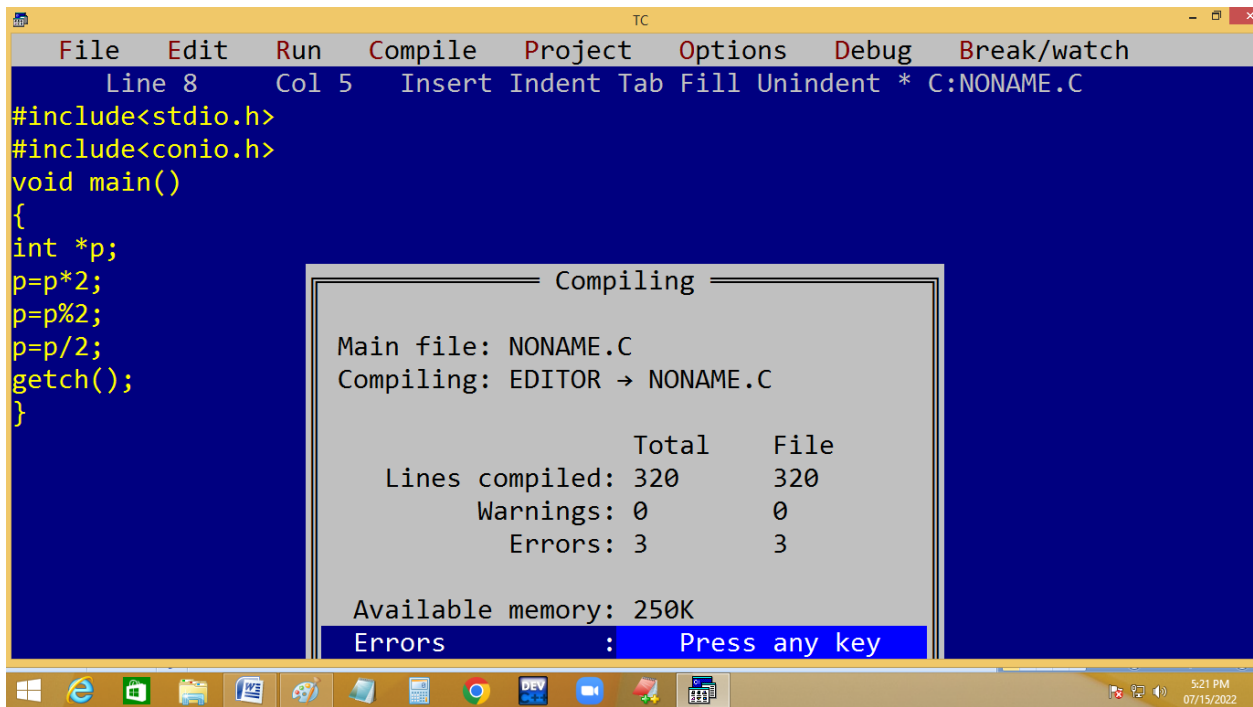
Like general variables we can compare, +, -, ++, -- the pointer values. But we can't perform %, / , * on pointers.



The screenshot shows a Turbo C++ (TC) IDE window. The title bar reads 'TC'. The menu bar includes 'File', 'Edit', 'Run', 'Compile', 'Project', 'Options', 'Debug', and 'Break/watch'. The status bar at the top indicates 'Line 6', 'Col 5', and 'Insert' mode. The main editing area has a dark blue background with yellow text. The code is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int *p;
p++;_
p--;
p+2;
p-2;
getch();
}
```

The Windows taskbar is visible at the bottom, showing icons for various applications and the system clock displaying 5:21 PM on 07/15/2022.



The screenshot shows the Turbo C++ IDE with a compilation window open. The main window displays the source code for NONAME.C, which includes `<stdio.h>` and `<conio.h>`, and defines a `main` function that declares an integer pointer `p`, performs arithmetic operations, and calls `getch()`. The compilation window, titled "Compiling", shows the main file as NONAME.C and the compiler as EDITOR. It provides a summary of the compilation: 320 lines compiled, 0 warnings, and 3 errors. The available memory is 250K. The window prompts the user to "Press any key".

```
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 5 Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int *p;
p=p*2;
p=p%2;
p=p/2;
getch();
}
```

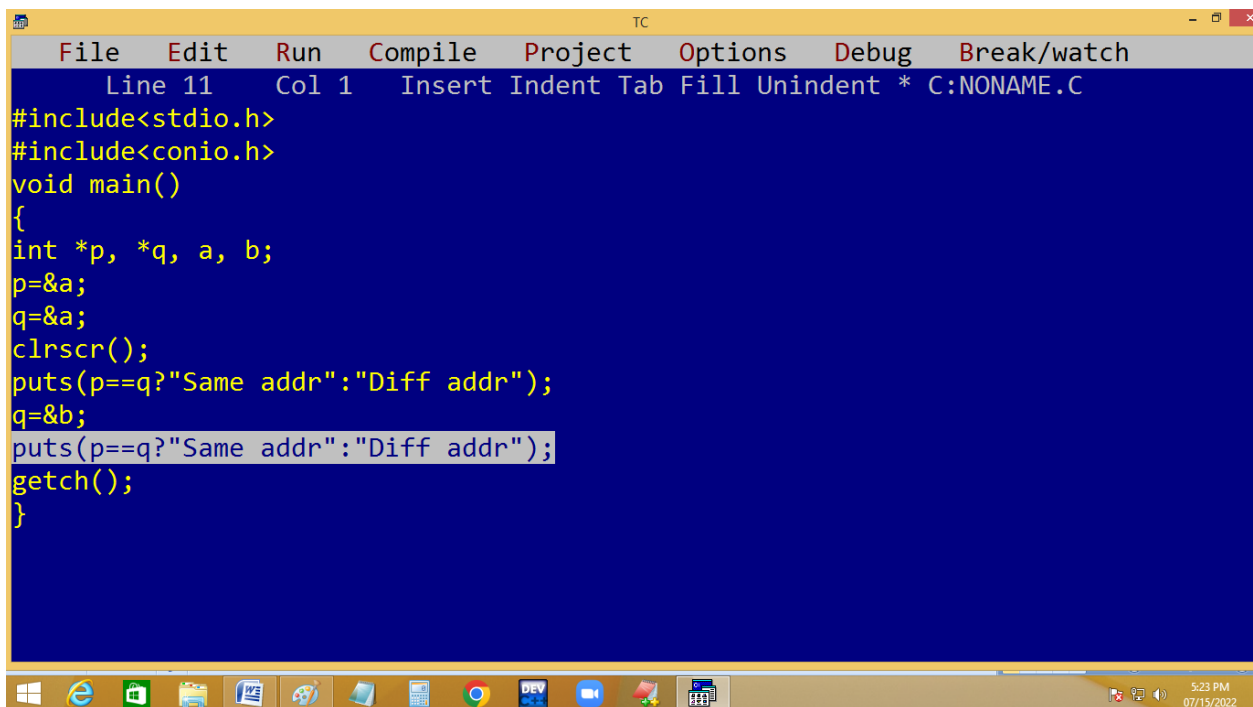
Compiling

Main file: NONAME.C
Compiling: EDITOR → NONAME.C

	Total	File
Lines compiled:	320	320
Warnings:	0	0
Errors:	3	3

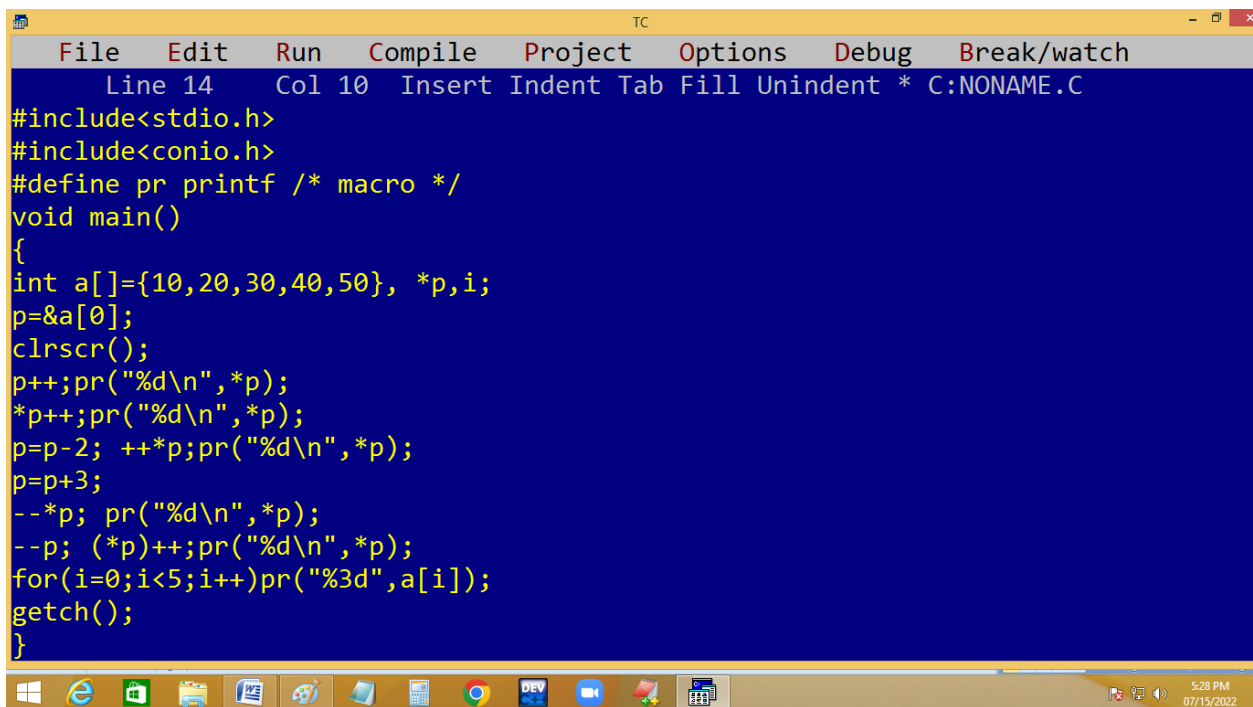
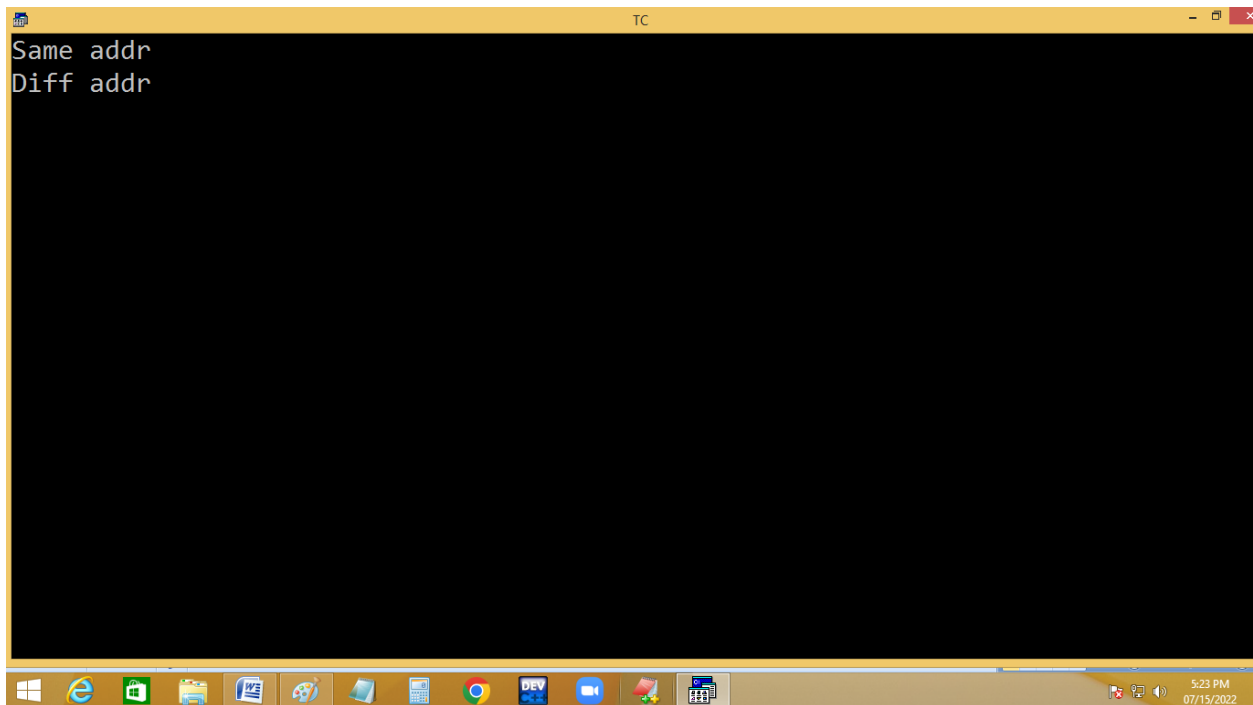
Available memory: 250K

Errors : Press any key



The screenshot shows the Turbo C++ IDE with the source code for NONAME.C. The code includes `<stdio.h>` and `<conio.h>`, and defines a `main` function that declares two integer pointers `p` and `q`, and two integers `a` and `b`. It assigns `p` and `q` the addresses of `a` and `b` respectively, clears the screen, and prints the results of `p==q` comparisons. The code is currently at line 11, column 1.

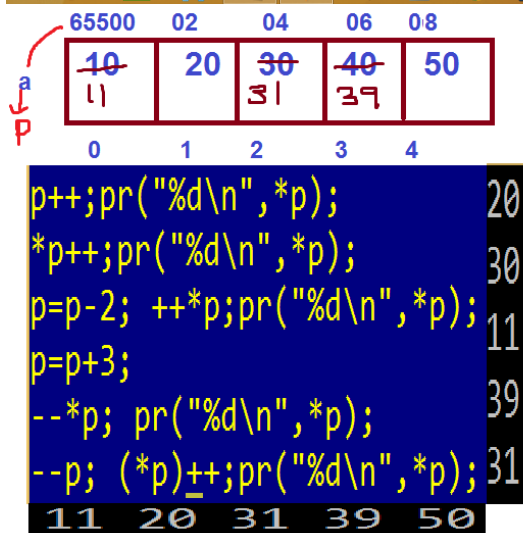
```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 1 Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
void main()
{
int *p, *q, a, b;
p=&a;
q=&a;
clrscr();
puts(p==q?"Same addr":"Diff addr");
q=&b;
puts(p==q?"Same addr":"Diff addr");
getch();
}
```



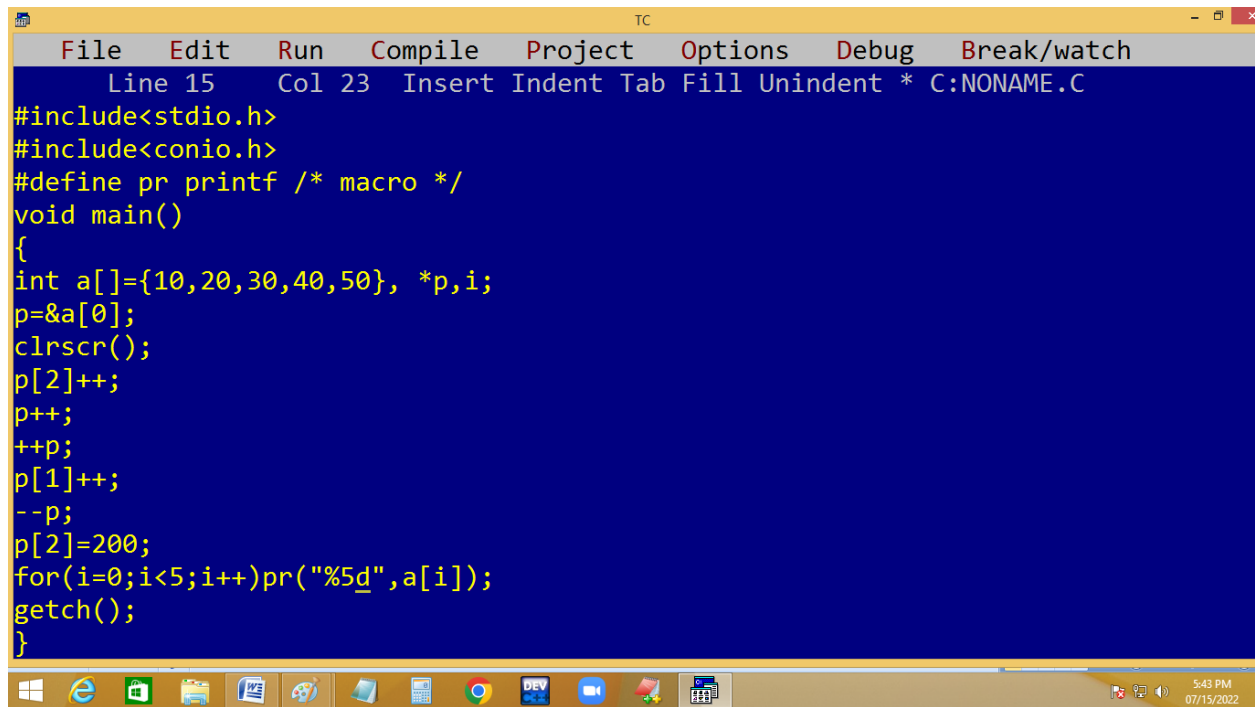
```

TC
20
30
11
39
31
11 20 31 39 50_

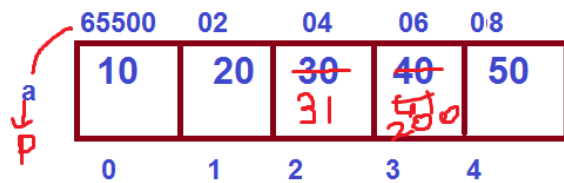
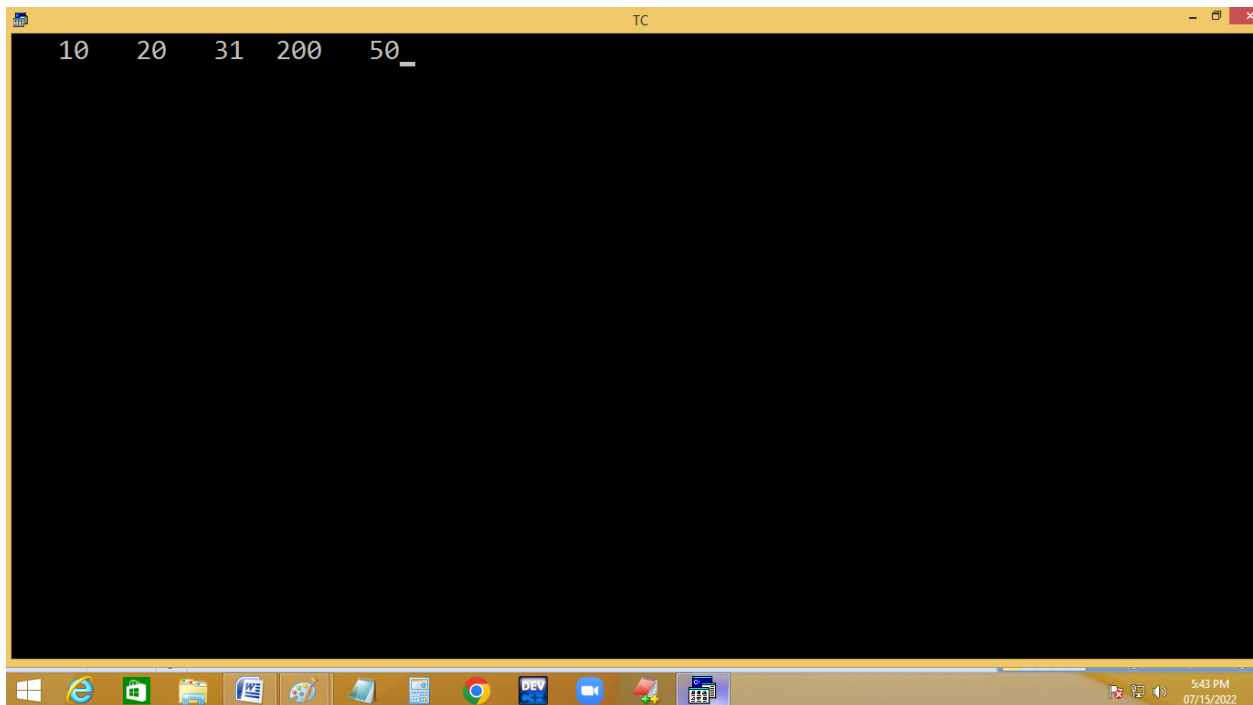
```



$p++ \Rightarrow 65500 + 1 * 2 = 65502;$
 $pr(*p) \Rightarrow \text{value at } 65502 \Rightarrow 20$ ✓
 $*p++ \Rightarrow 65502 + 1 * 2 = 65504$
 $pr(*p) \Rightarrow \text{value at } 65504 \Rightarrow 30$ ✓
 $p = p - 2 \Rightarrow 65504 - 2 * 2 \Rightarrow 65500$
 $++*p \Rightarrow \text{value at } 65500 \Rightarrow 10 \Rightarrow 11$
 $pr(*p) \Rightarrow \text{value at } 65500 \Rightarrow 11$ ✓
 $p = p + 3 \Rightarrow 65500 + 3 * 2 \Rightarrow 65506$
 $--*p \Rightarrow \text{value at } 65506 \Rightarrow 40 \Rightarrow 39$
 $pr(*p) \Rightarrow \text{value at } 65506 \Rightarrow 39$ ✓
 $--p \Rightarrow 65506 - 1 * 2 \Rightarrow 65504$
 $(*p)++ \Rightarrow \text{value at } 65504 \Rightarrow 30 \Rightarrow 31$
 $pr(*p) \Rightarrow \text{value at } 65504 \Rightarrow 31$ ✓



```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 15 Col 23 Insert Indent Tab Fill Unindent * C:NONAME.C
#include<stdio.h>
#include<conio.h>
#define pr printf /* macro */
void main()
{
int a[]={10,20,30,40,50}, *p,i;
p=&a[0];
clrscr();
p[2]++;
p++;
++p;
p[1]++;
--p;
p[2]=200;
for(i=0;i<5;i++)pr("%5d",a[i]);
getch();
}
```



```

p[2]++;
p++;
++p;
p[1]++;
--p;
p[2]=200;

```

10 20 31 200 50

```

p = 65500
p[2]++==>30++ ==>31
p = 65500
p++==>65502
++p==>65504
p[1]++==> 65506++ ==>40++ ==>41
p=65504
--p; 65502
p[2]=200;
65502+2*2==>65506==> 41 ==> 200

```