Pointer to array:

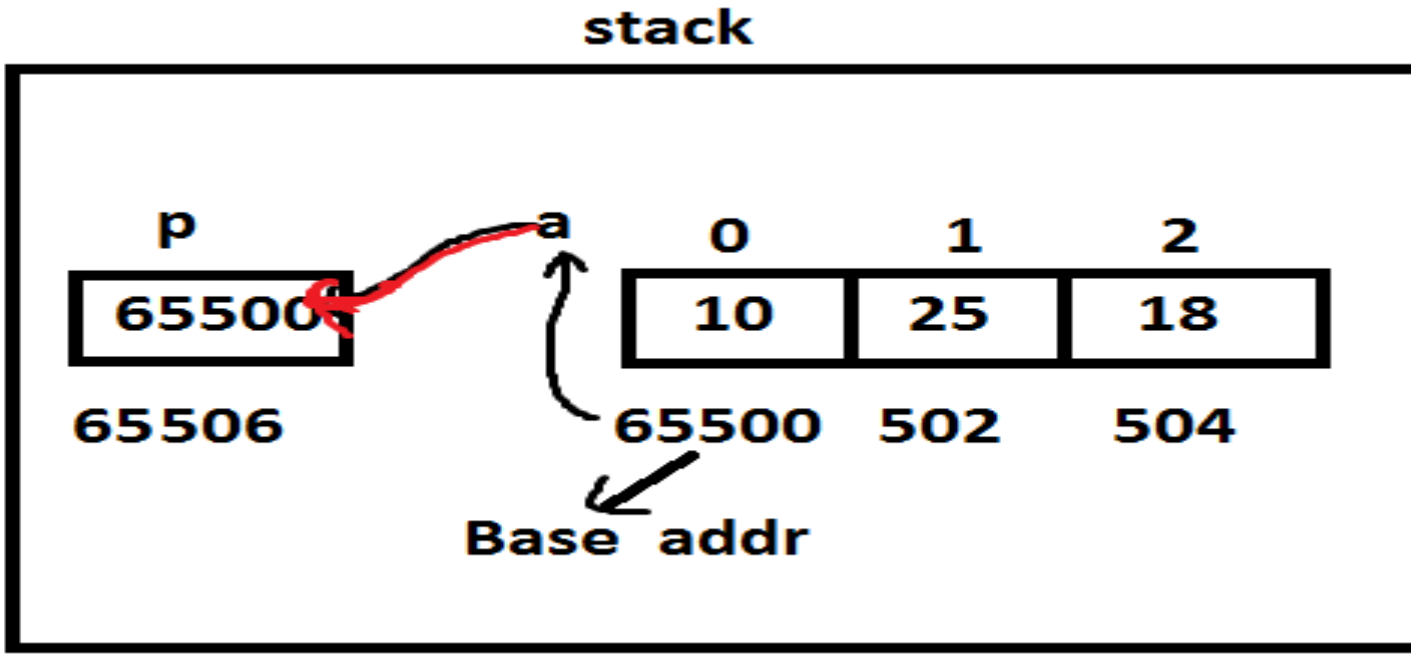
Array is implicit pointer. Due to this it holds the base cell addr [0 cell addr] implicitly. By assigning the array name or 0 cell addr to the pointer, we can handle array elements using the following syntax.

***(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.

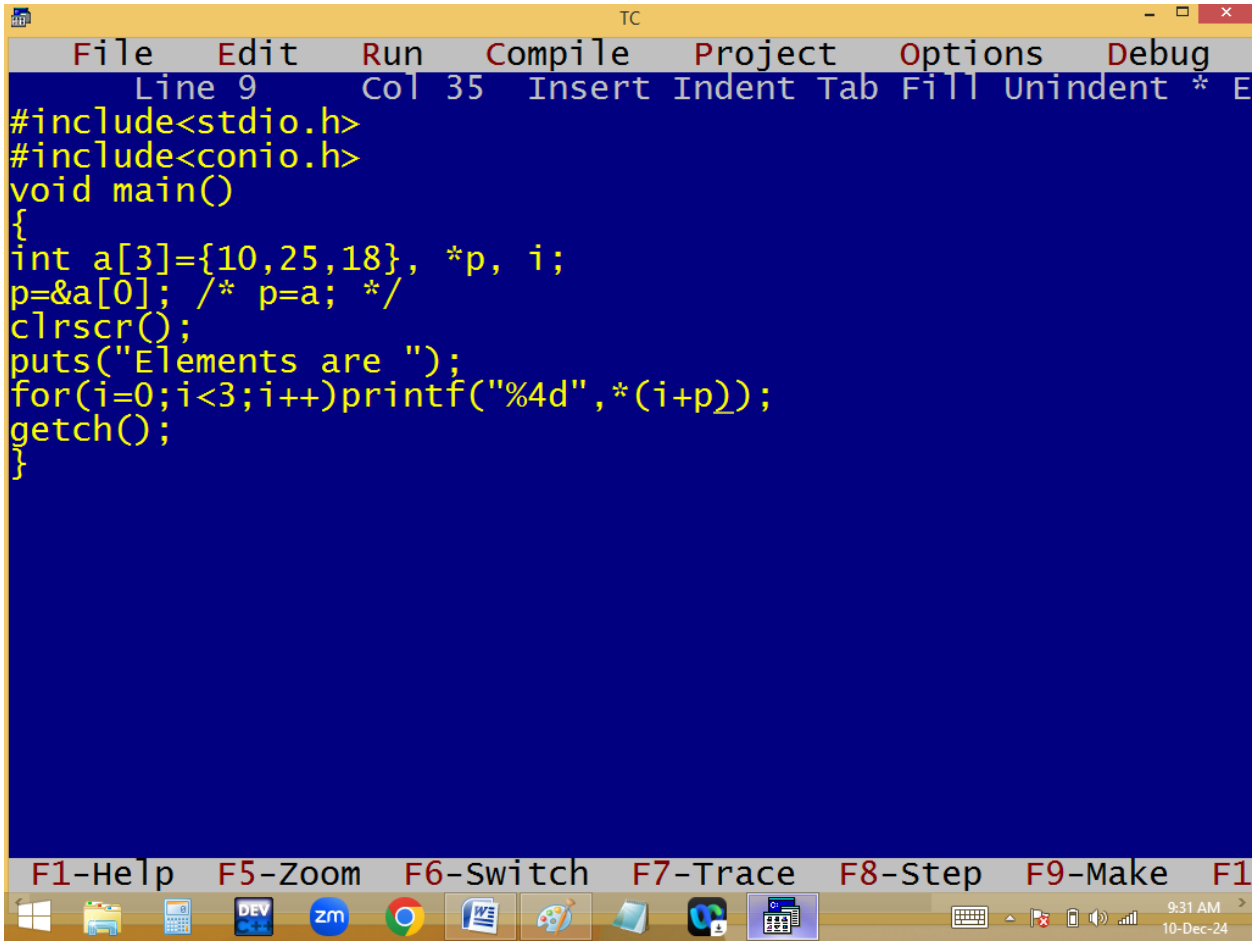
Eg:

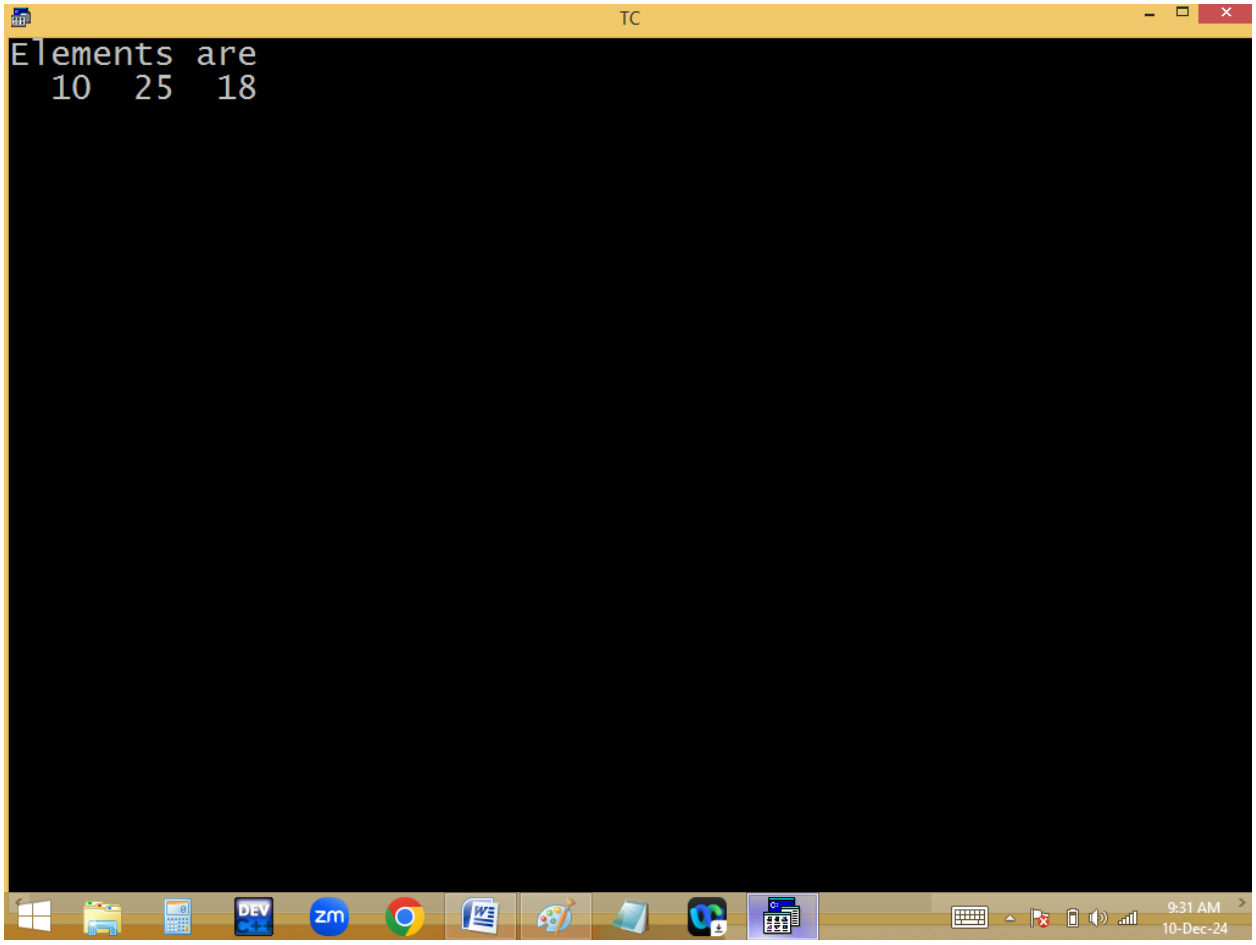
```
#include<stdio.h>
#include<conio.h>
void  main()
{
int  *p,  a[3]={10,25,18}, i;
clrscr();
p = a;   /* p=&a[0]; or p=&a; */
printf("Elements are: ");
for(i=0;i<3;i++)
printf("%4d",*(p+i));
getch();
}
```

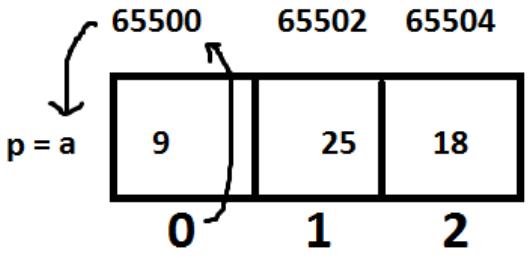
Output: Elements are: 10 25 18

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
Line 9 Col 35 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a[3]={10,25,18}, *p, i;
p=&a[0]; /* p=a; */
clrscr();
puts("Elements are ");
for(i=0;i<3;i++)printf("%4d",*(i+p));
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:31 AM
10-Dec-24

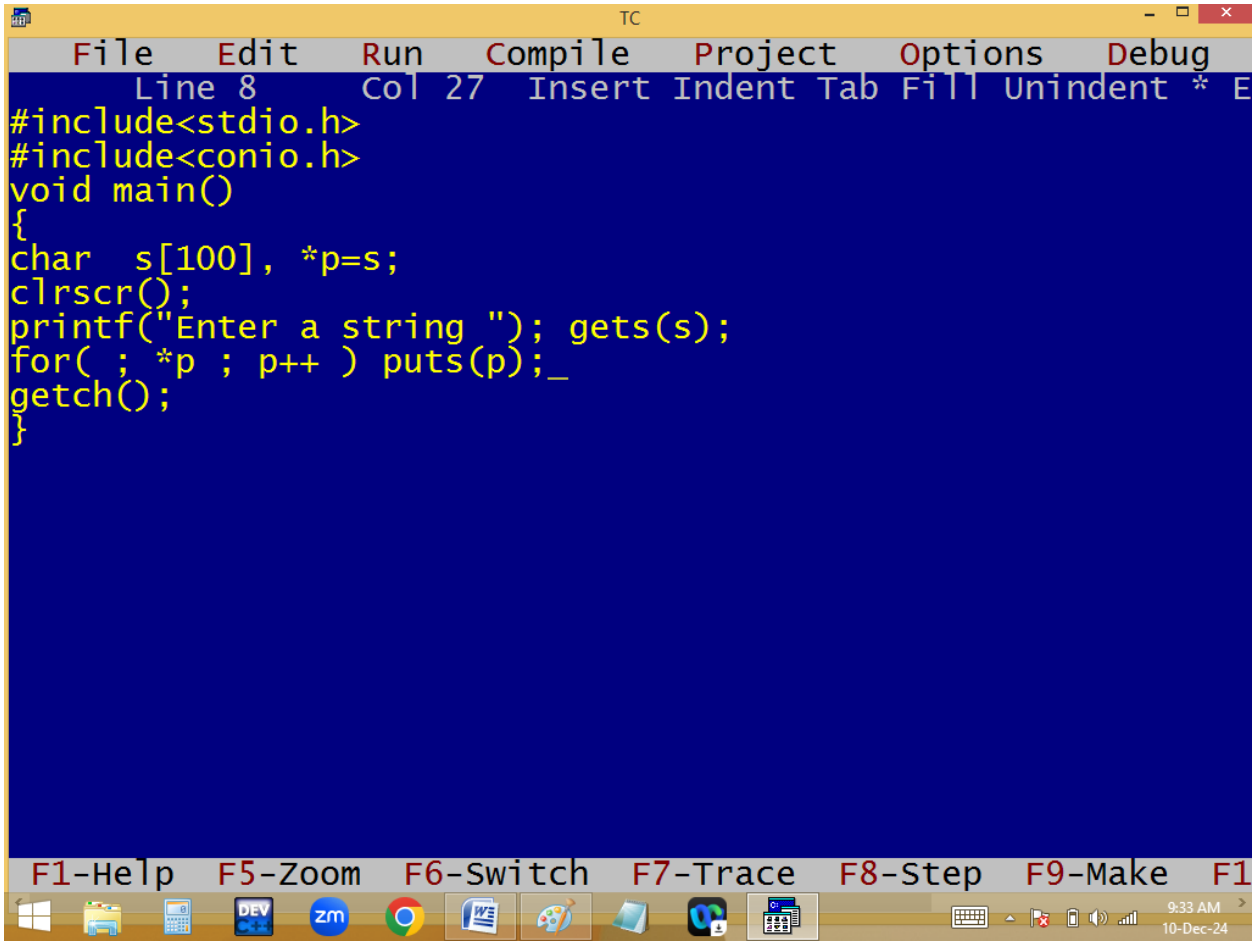


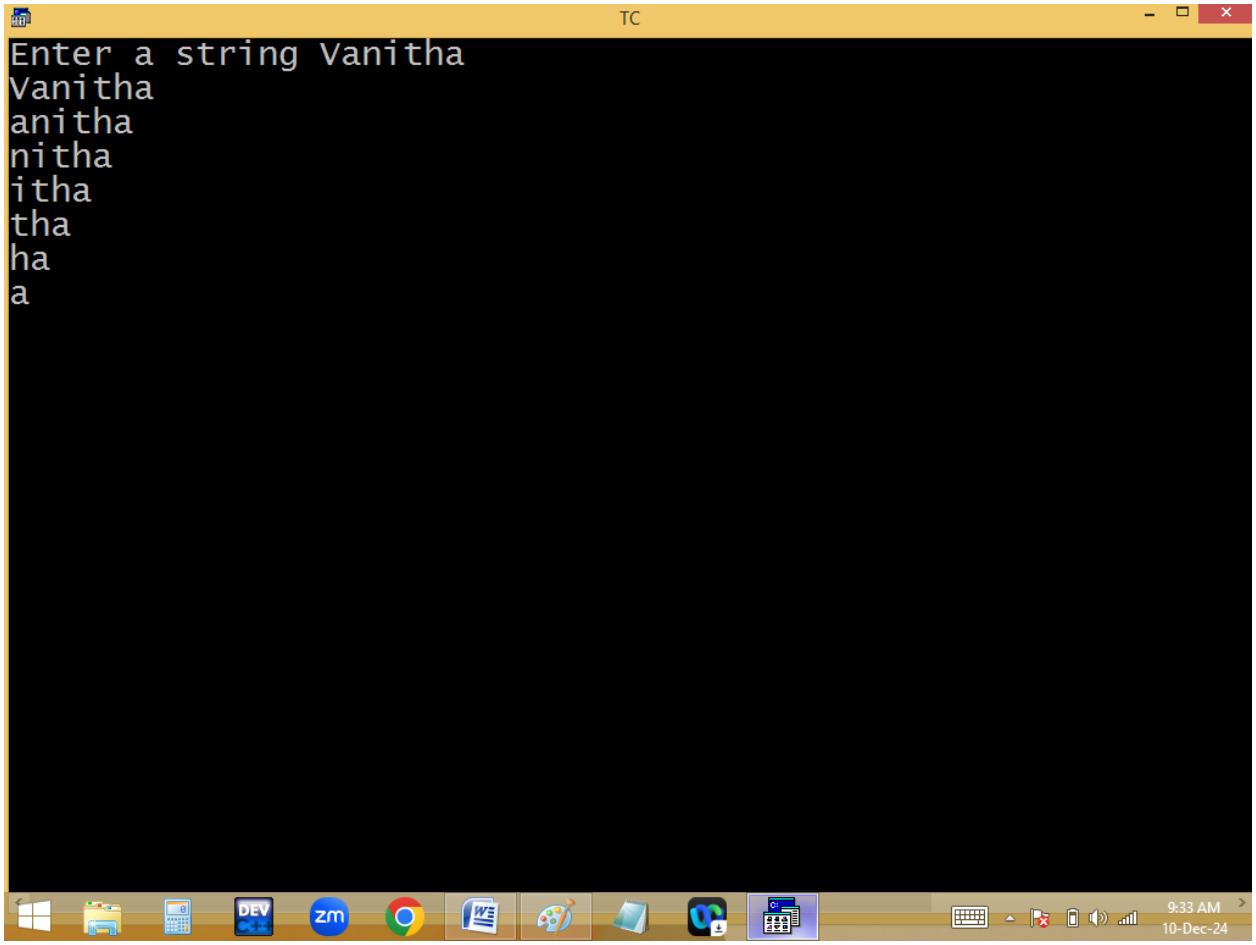


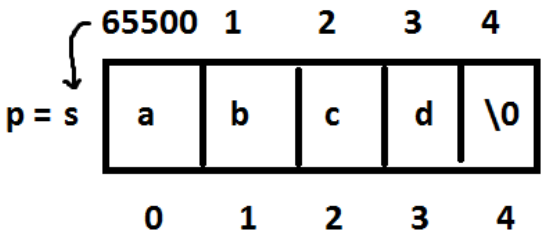
for(i=0; i<3;i++) p(* (p + i));

***65500+0*2=*65500==>value at 65500 ==> 9**
***65500+1*2=*65502==>value at 65502 ==> 25**
***65500+2*2=*65504==>value at 65504 ==> 18**

Pointer to string:


TC
File Edit Run Compile Project Options Debug
Line 8 Col 27 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p=s;
clrscr();
printf("Enter a string "); gets(s);
for(; *p ; p++) puts(p);_
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:33 AM
10-Dec-24





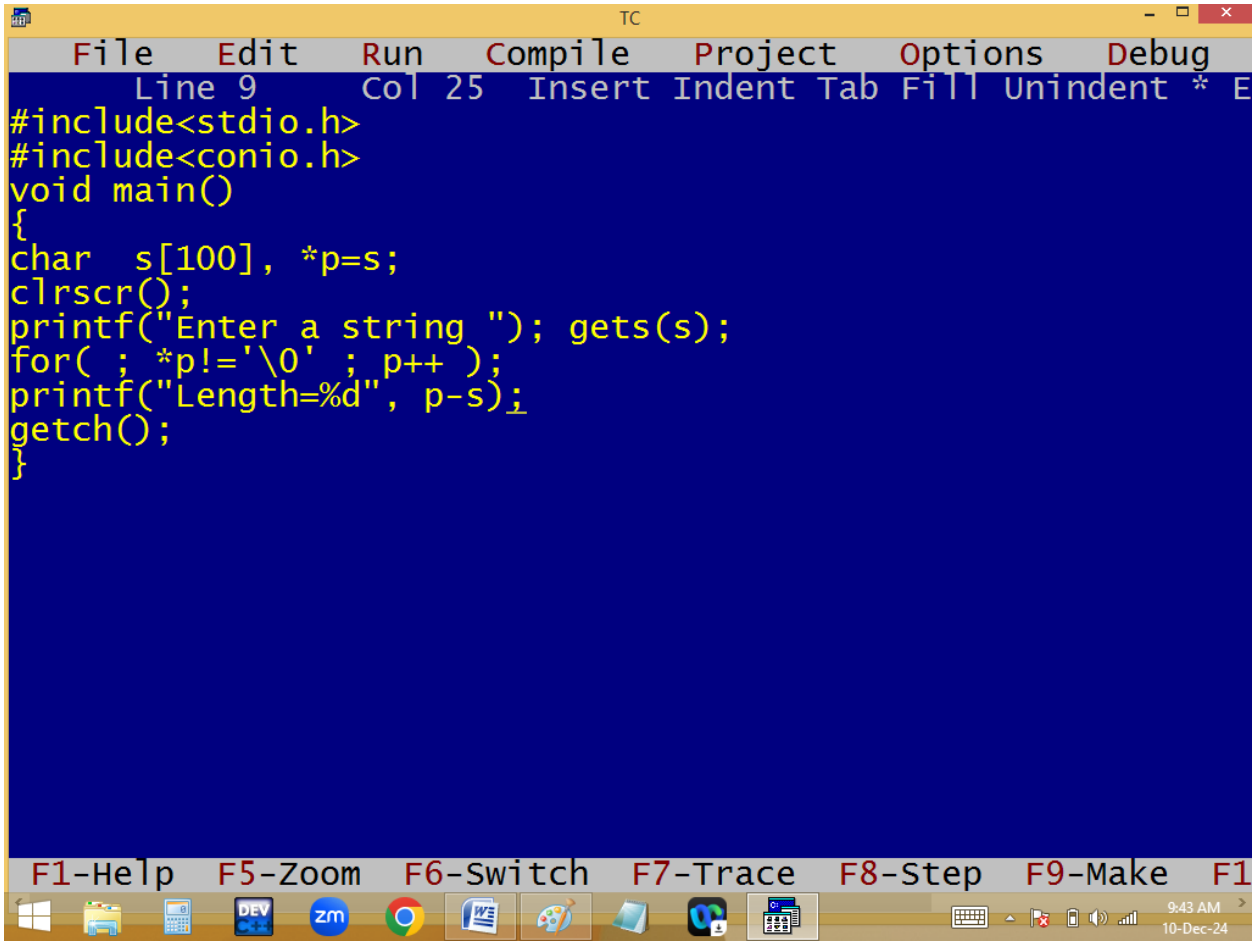
*p!='\0'

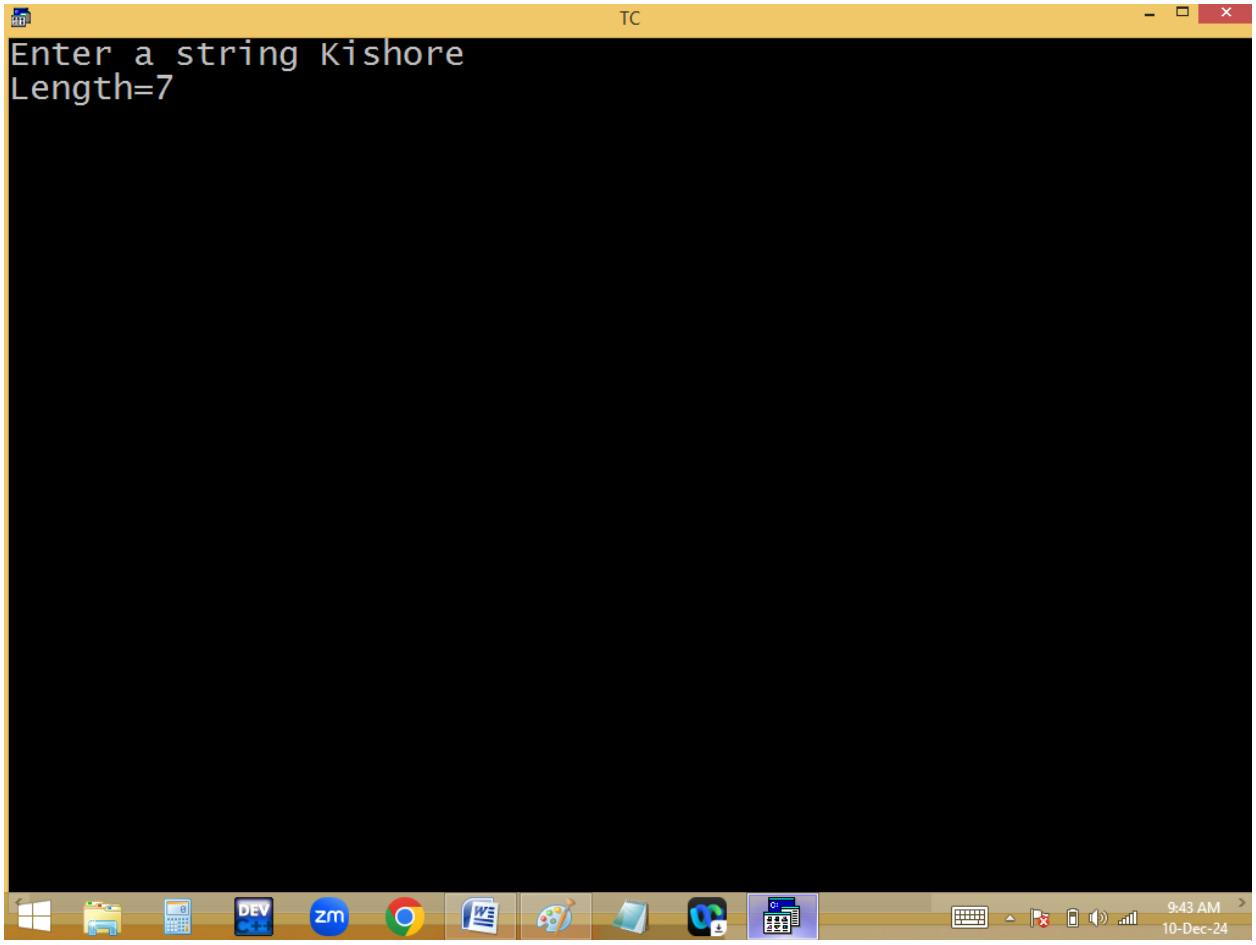
for(; *p ; p++) puts (p);

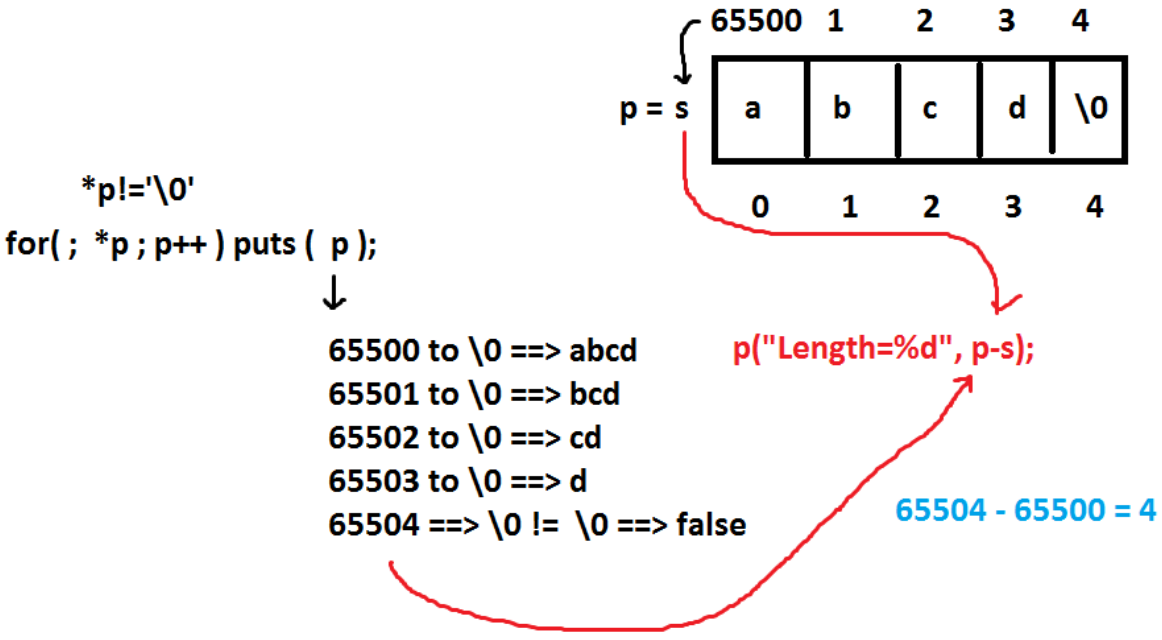
↓

65500 to \0 ==> abcd
65501 to \0 ==> bcd
65502 to \0 ==> cd
65503 to \0 ==> d
65504 ==> \0 != \0 ==> false

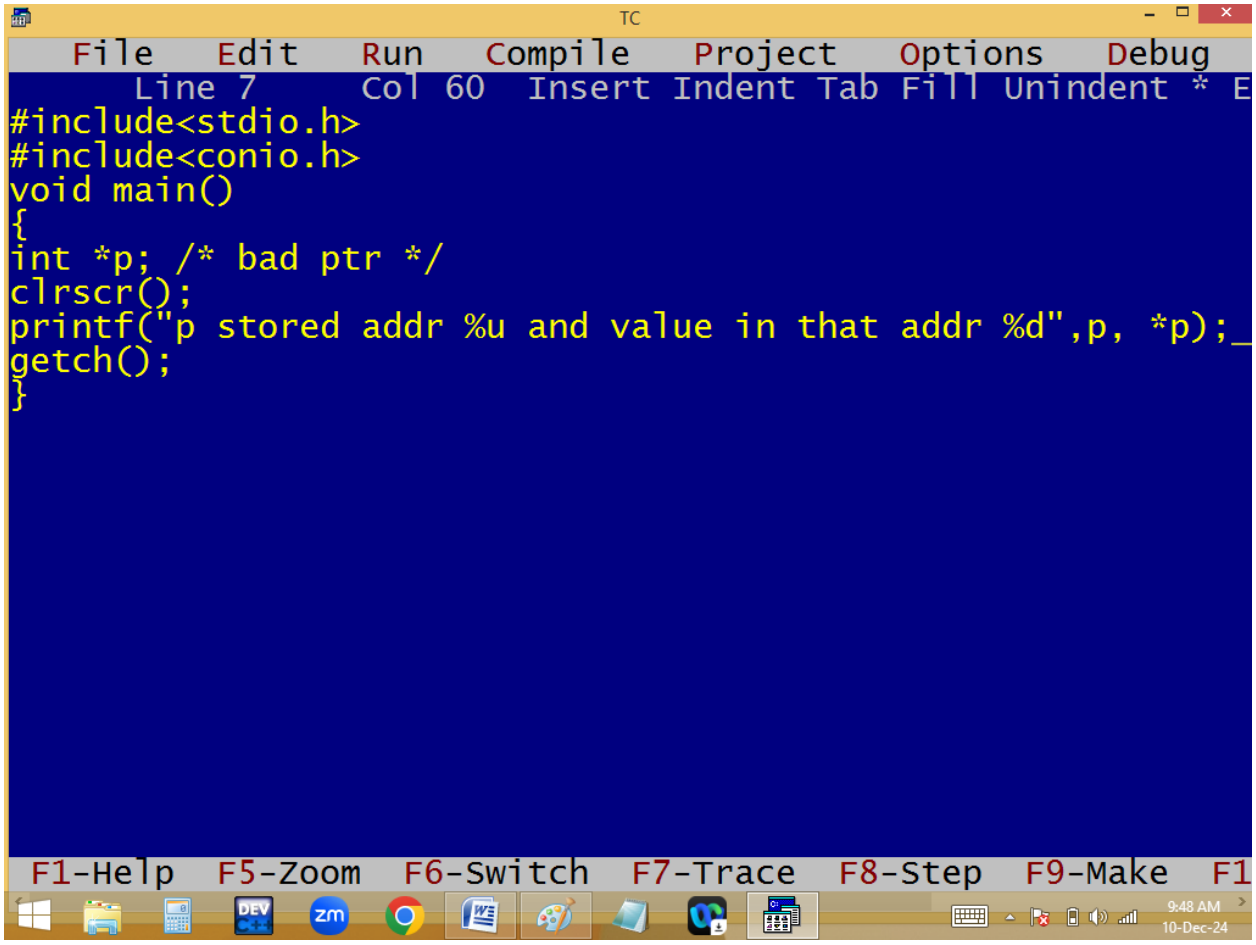
Finding string length using pointer only:

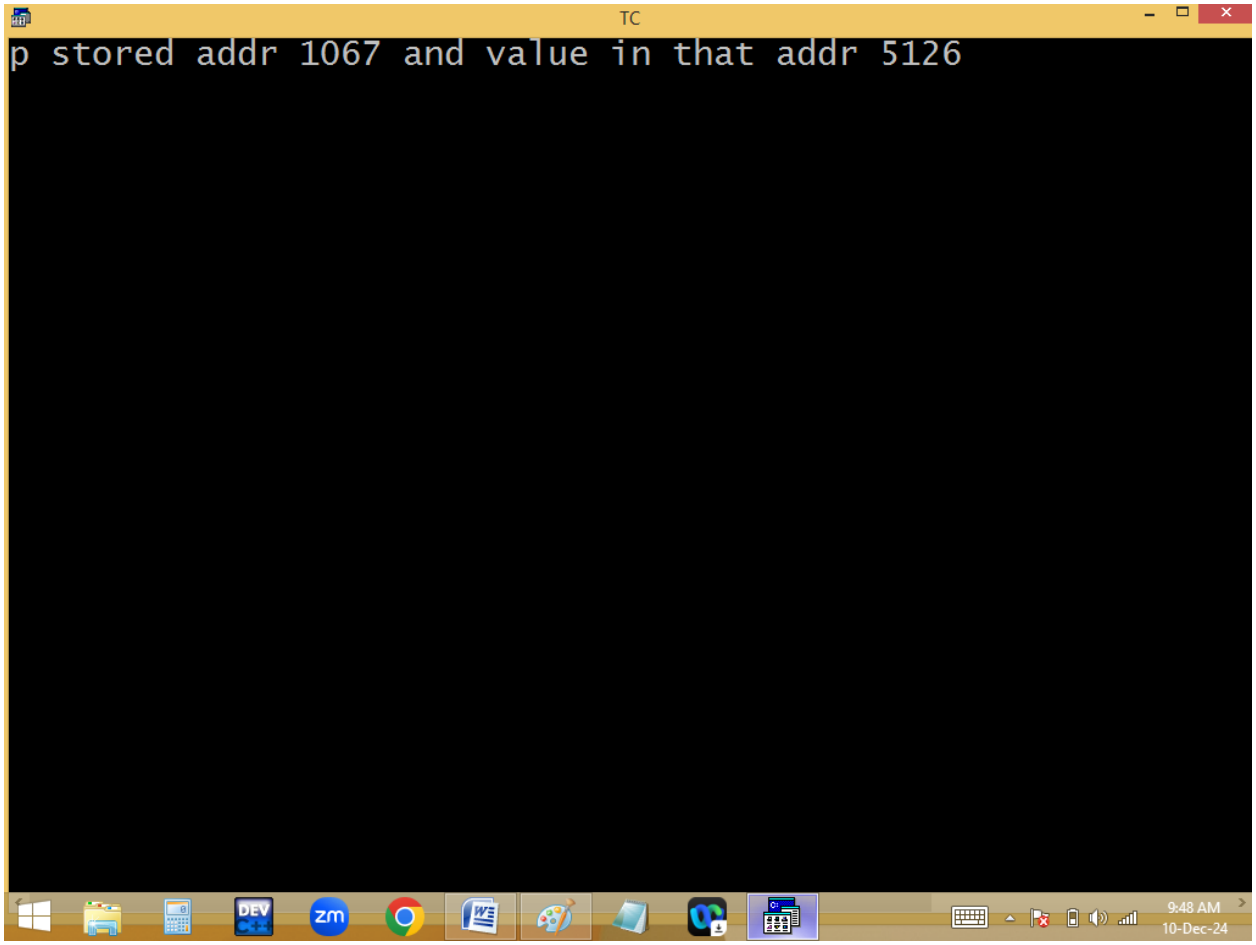

TC
File Edit Run Compile Project Options Debug
Line 9 Col 25 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p=s;
clrscr();
printf("Enter a string "); gets(s);
for(; *p!='\0' ; p++);
printf("Length=%d", p-s);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:43 AM
10-Dec-24


TC
Enter a string Kishore
Length=7
9:43 AM
10-Dec-24

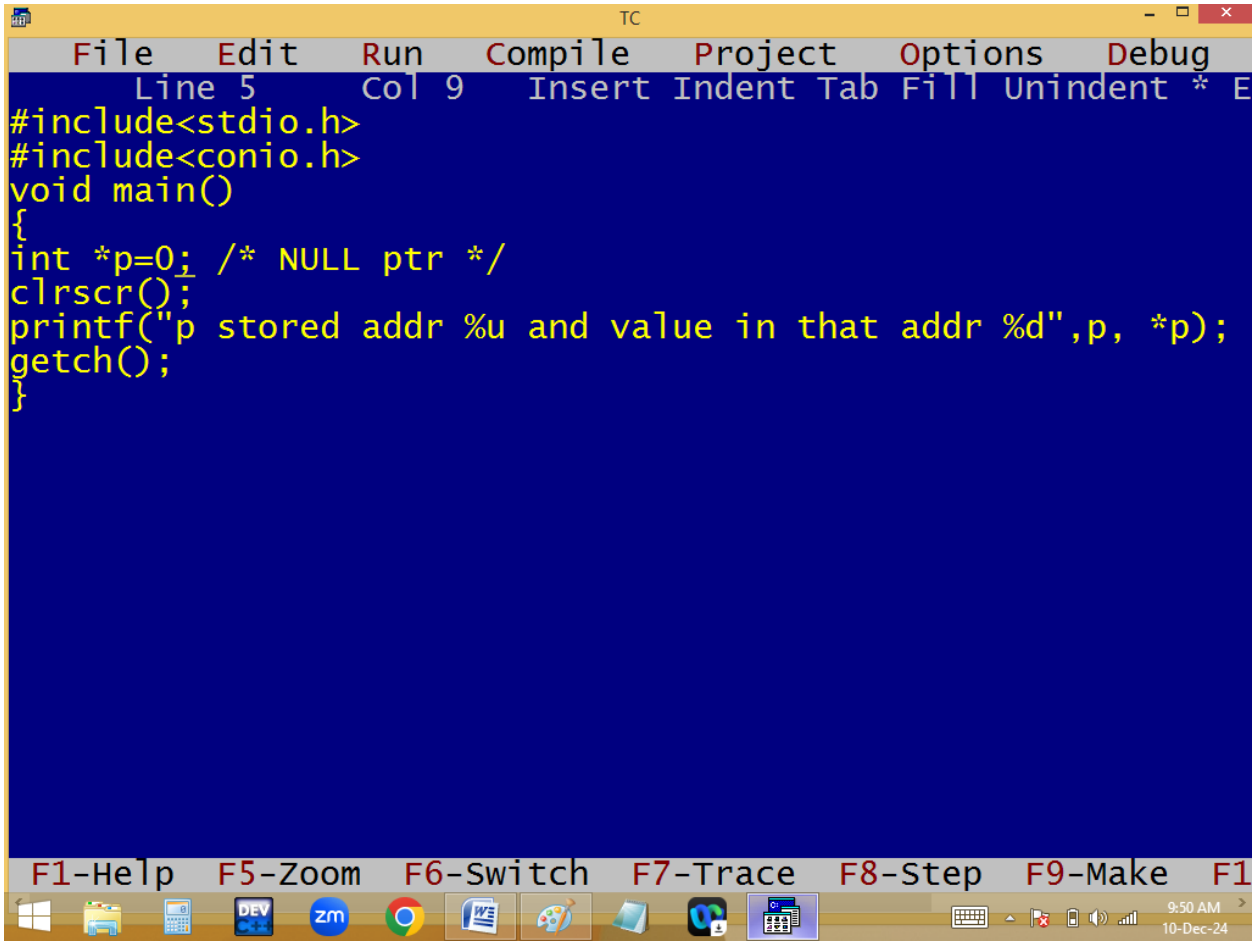

65500 1 2 3 4
p = s
a b c d \0
0 1 2 3 4
*p!='\0'
for(; *p ; p++) puts (p);
65500 to \0 ==> abcd
65501 to \0 ==> bcd
65502 to \0 ==> cd
65503 to \0 ==> d
65504 ==> \0 != \0 ==> false
p("Length=%d", p-s);
65504 - 65500 = 4

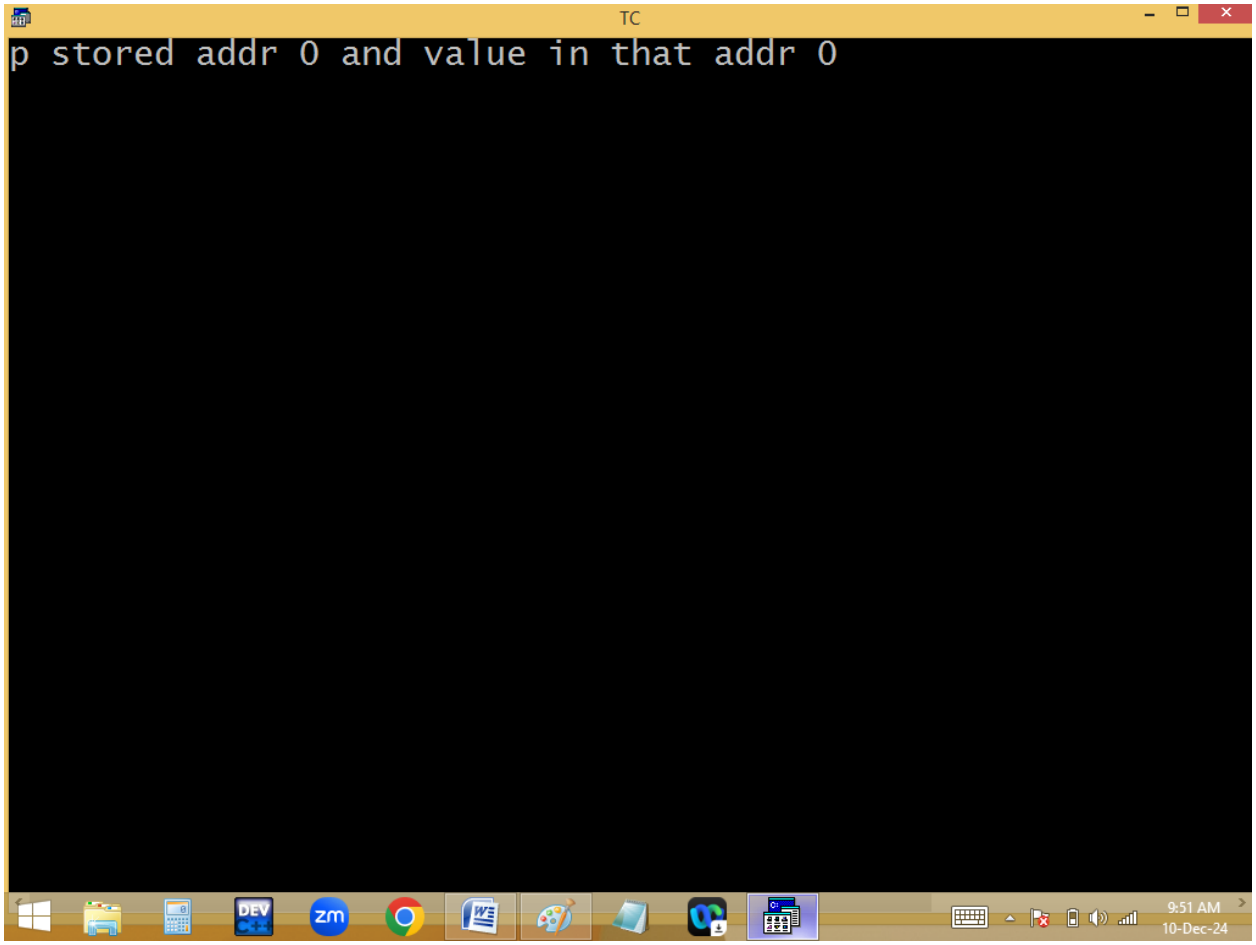
Bad / wild pointer: A pointer is declared but not initialized. In this situation the pointer is storing some unknown value and address. This kind of pointer is called bad / wild pointer.

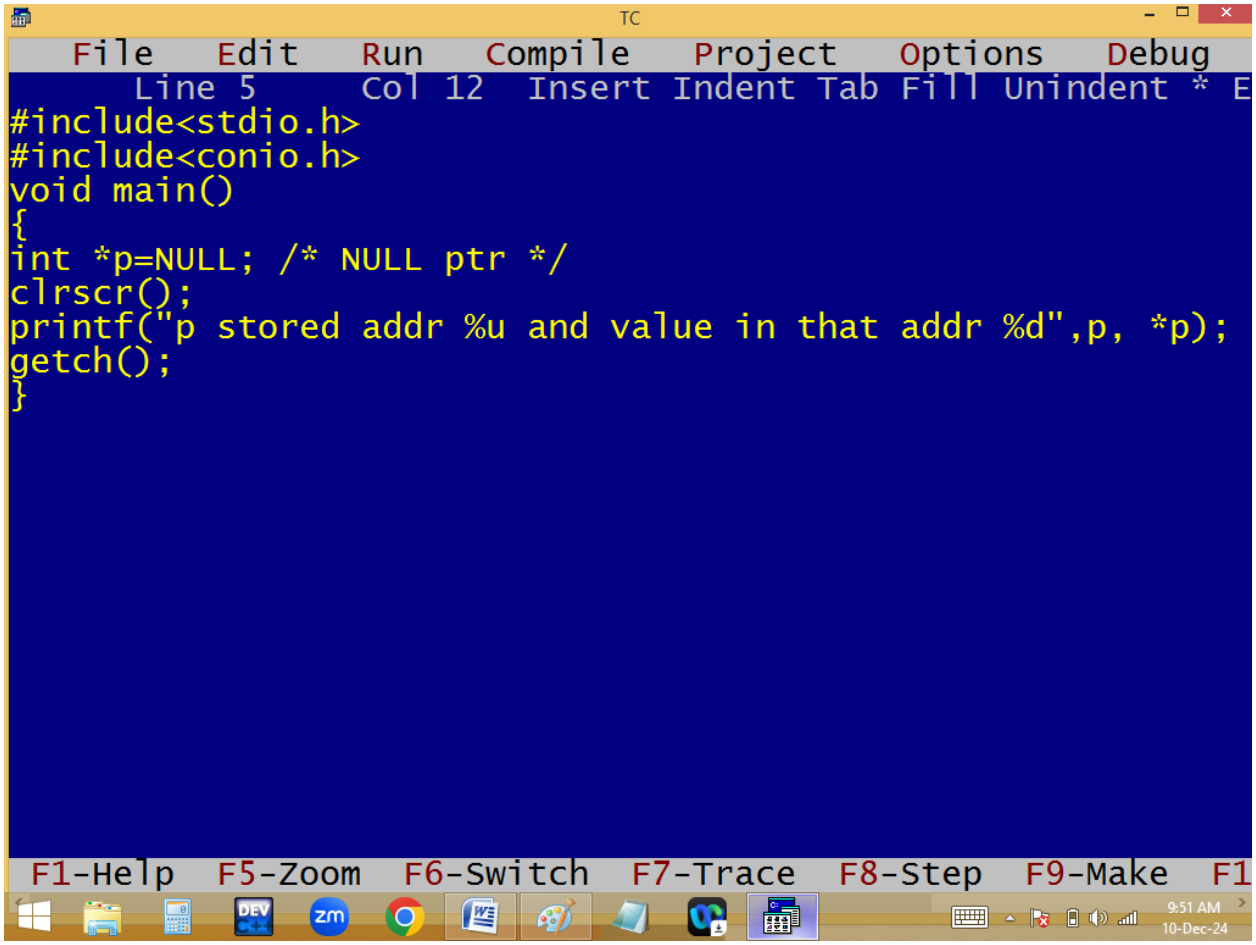


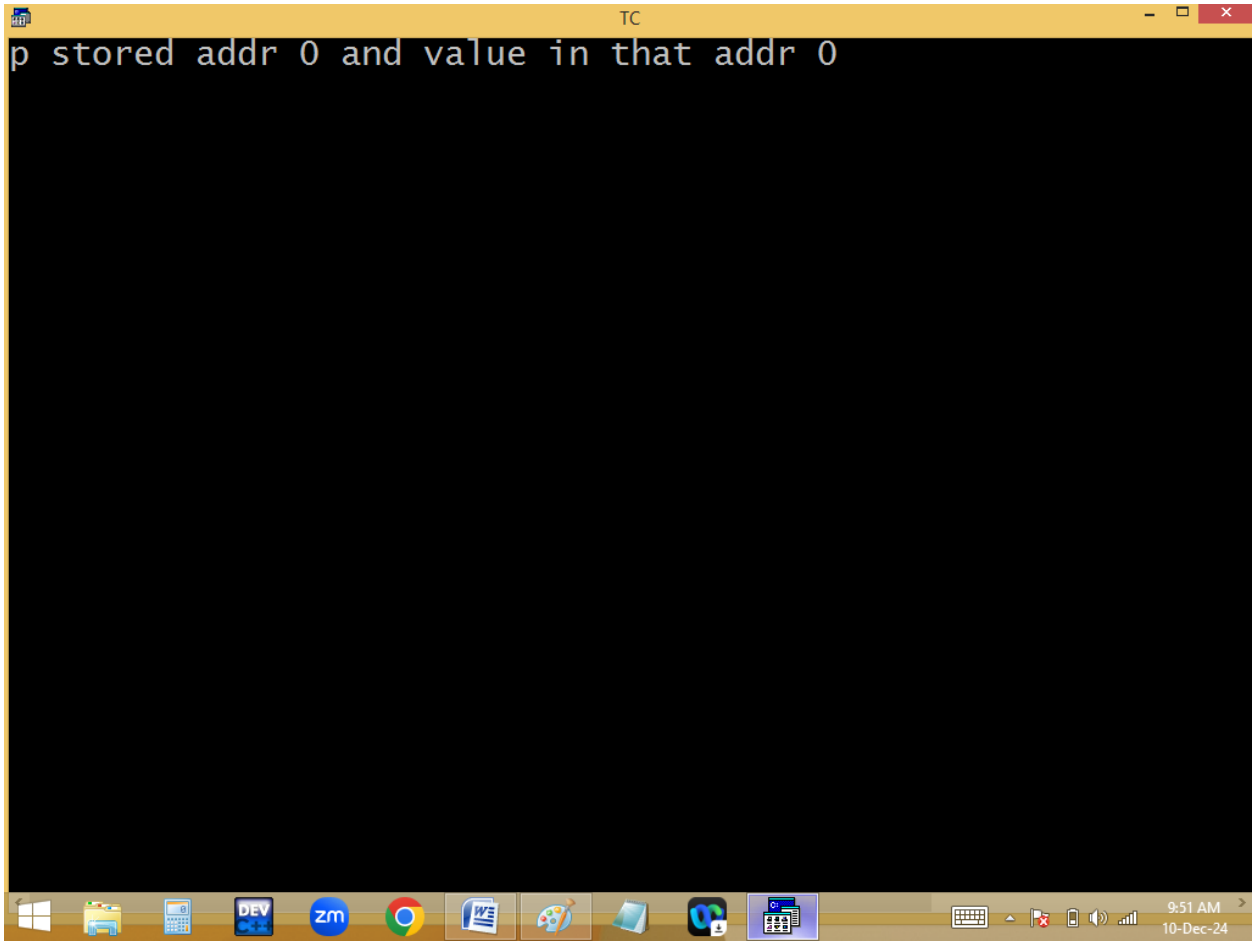


NULL pointer: When a pointer initialized with 0 / NULL then it is a NULL pointer. To avoid bad and dangling pointers we are using NULL pointer.

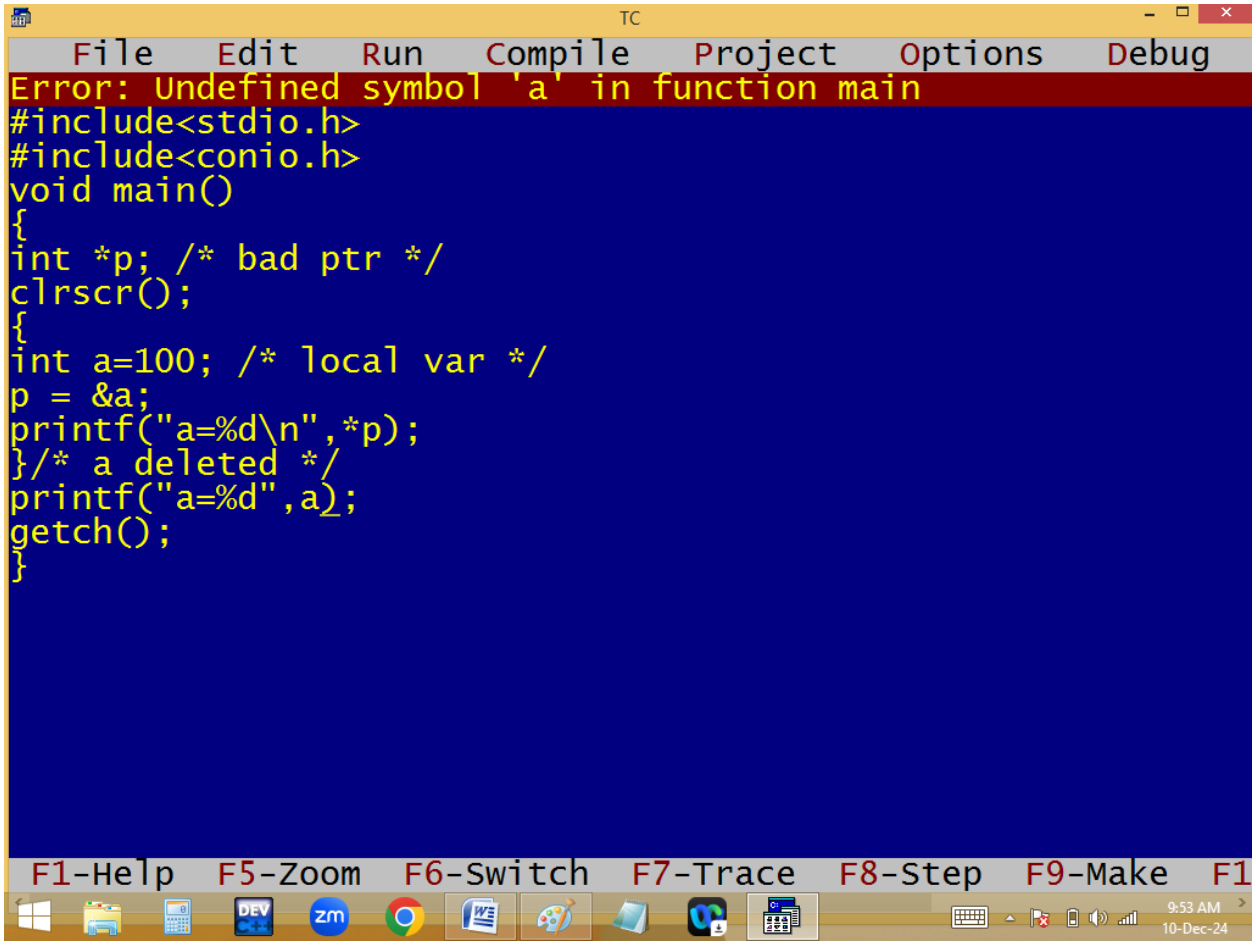

TC
File Edit Run Compile Project Options Debug
Line 5 Col 9 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int *p=0; /* NULL ptr */
clrscr();
printf("p stored addr %u and value in that addr %d",p, *p);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:50 AM
10-Dec-24

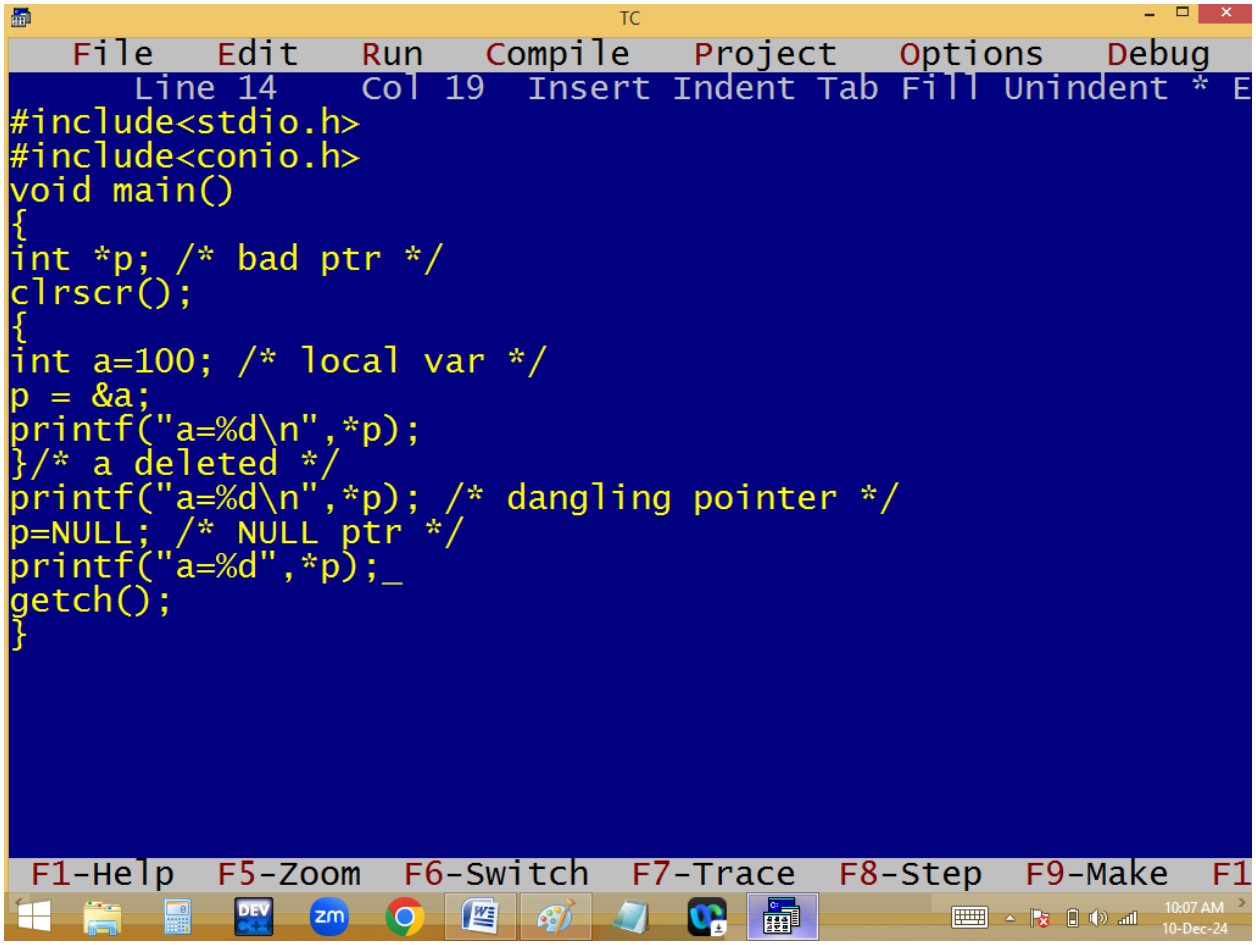

TC
p stored addr 0 and value in that addr 0
9:51 AM
10-Dec-24

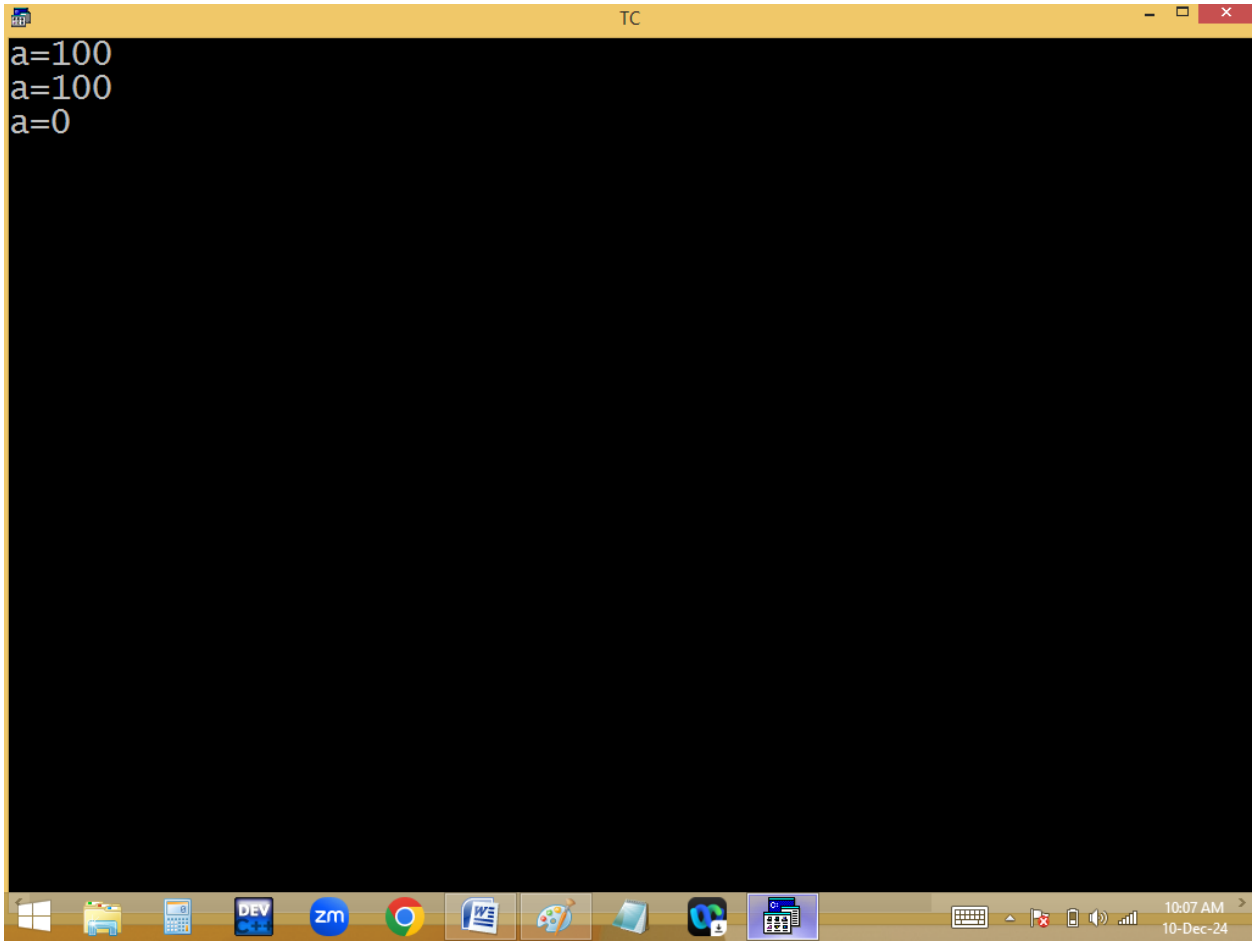

TC
File Edit Run Compile Project Options Debug
Line 5 Col 12 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int *p=NULL; /* NULL ptr */
clrscr();
printf("p stored addr %u and value in that addr %d",p, *p);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:51 AM
10-Dec-24



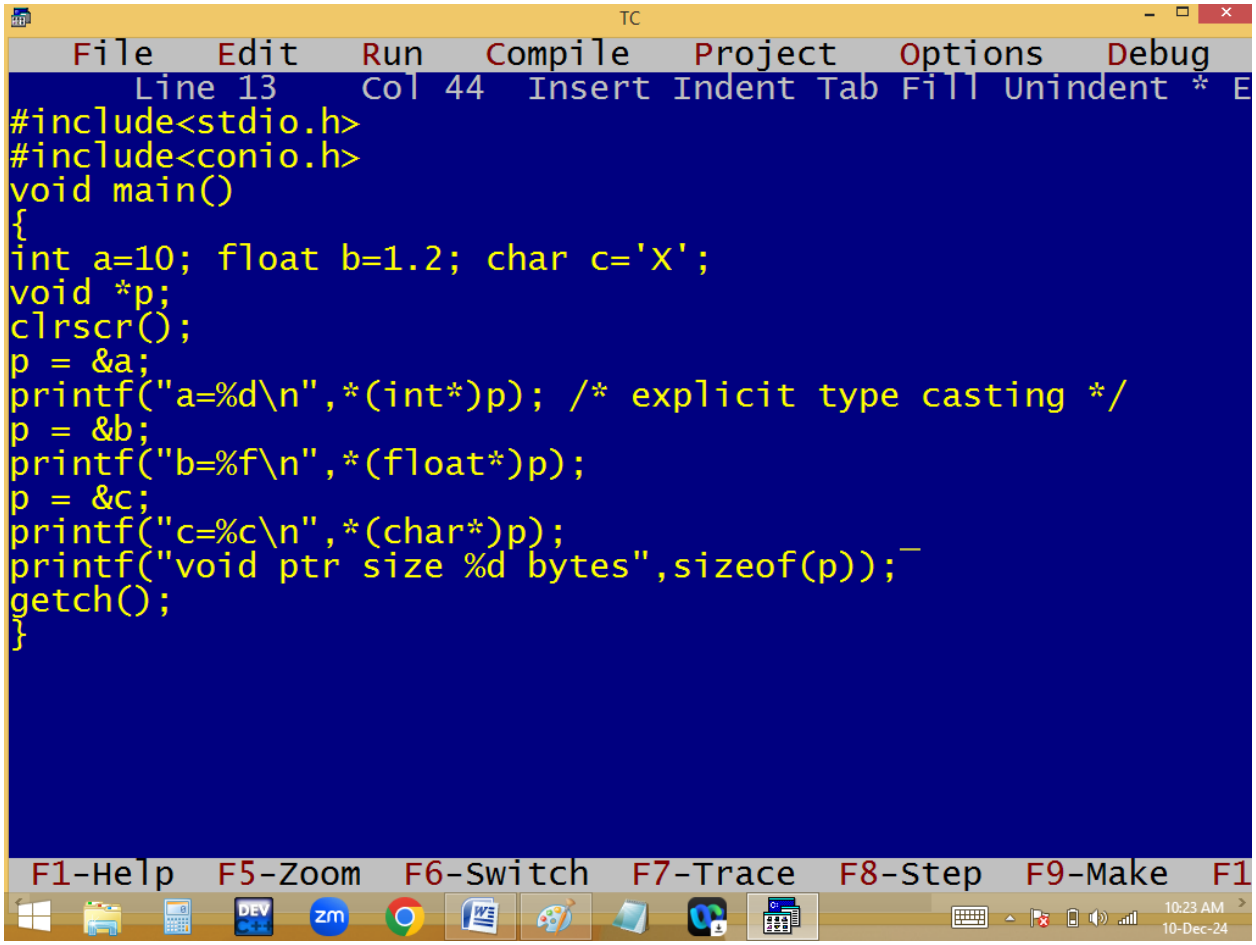
Dangling pointer: **A pointer is declared and some variable address also assigned. After some time that variable deleted from memory. But still the pointer is storing that deleted variable value and address. This kind of pointers are called dangling pointer and to avoid this initialize with NULL pointer.**

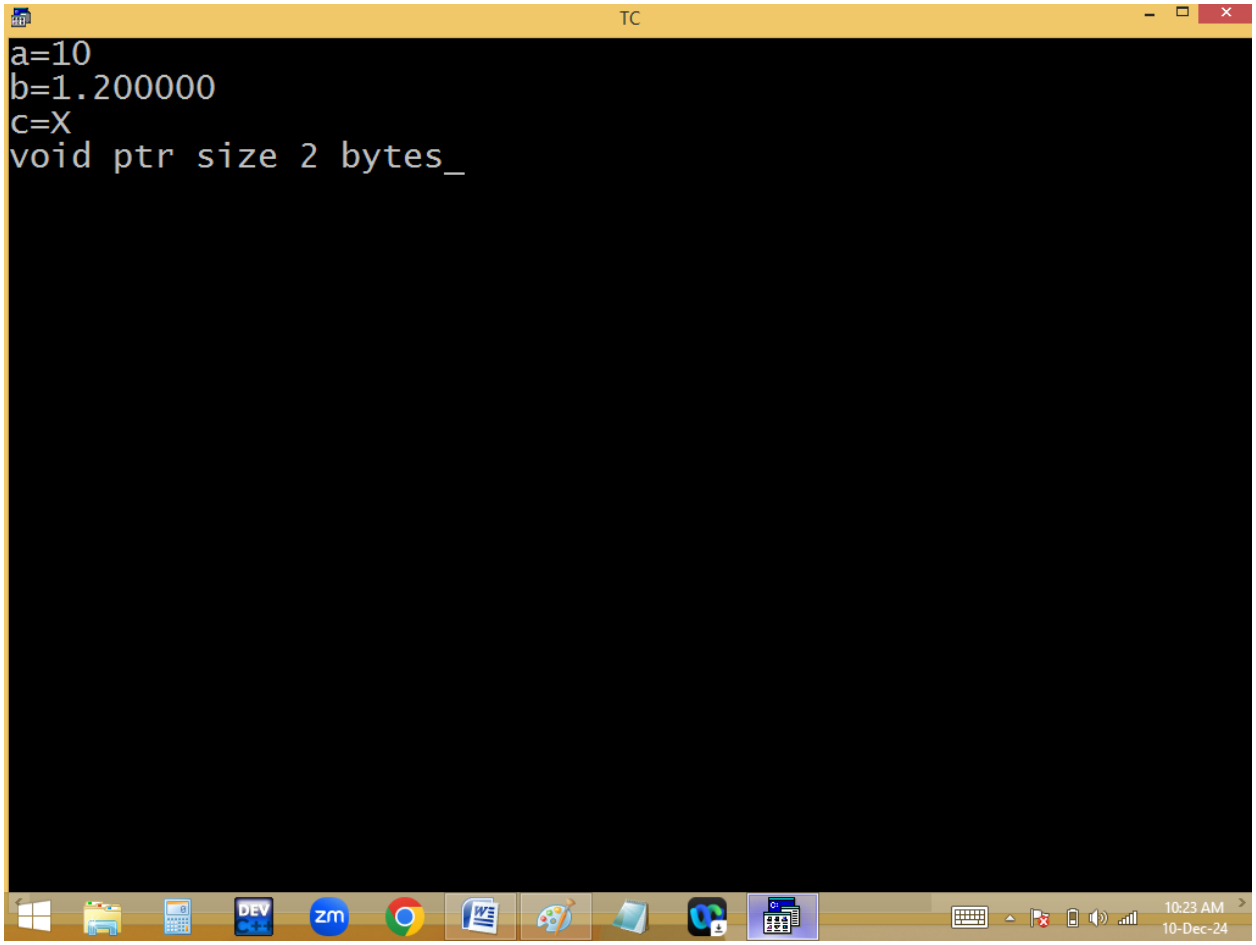

TC
File Edit Run Compile Project Options Debug
Error: Undefined symbol 'a' in function main
#include<stdio.h>
#include<conio.h>
void main()
{
int *p; /* bad ptr */
clrscr();
{
int a=100; /* local var */
p = &a;
printf("a=%d\n",*p);
}/* a deleted */
printf("a=%d",a);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
9:53 AM
10-Dec-24


TC
File Edit Run Compile Project Options Debug
Line 14 Col 19 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int *p; /* bad ptr */
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();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:07 AM
10-Dec-24



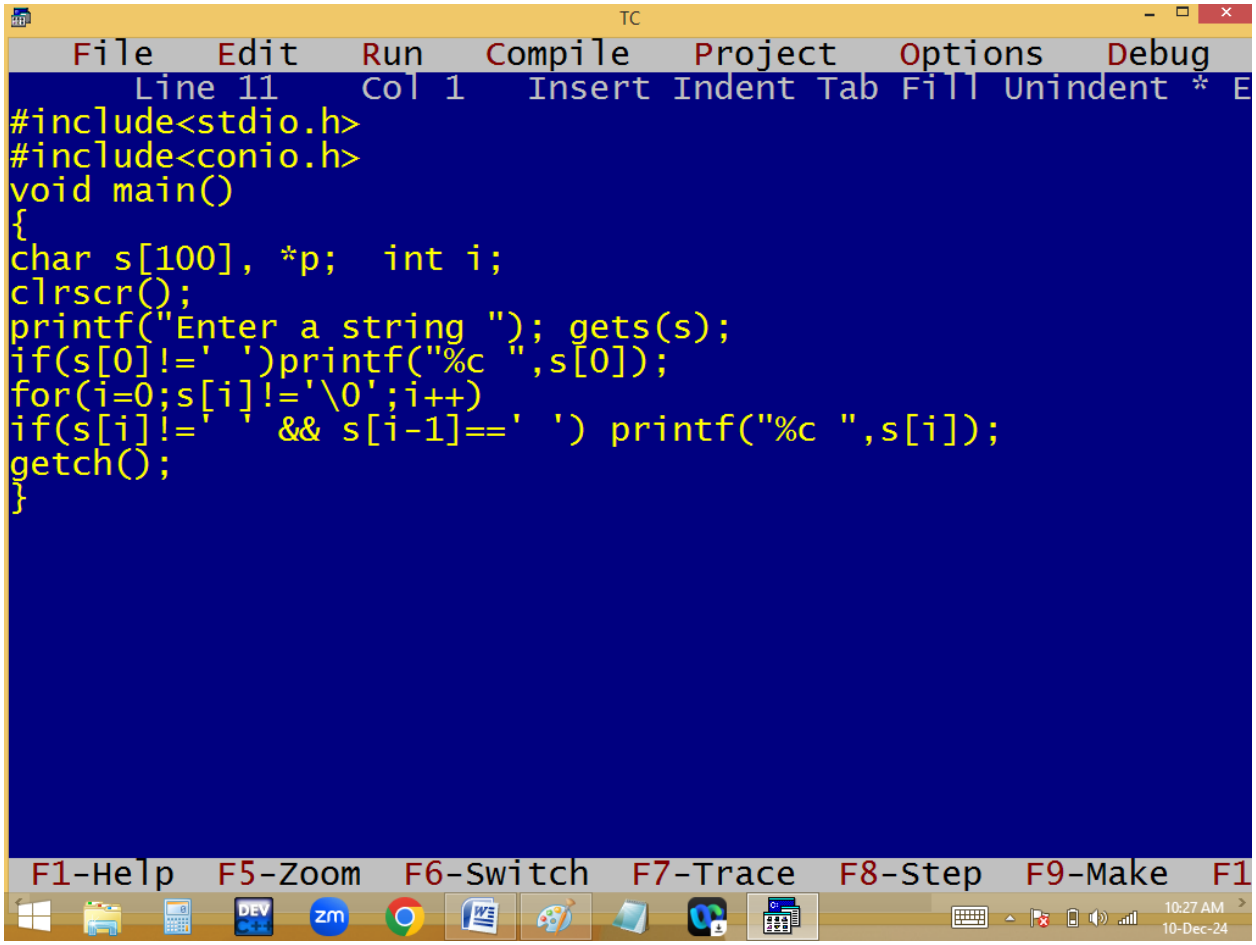
void / generic pointer: Pointer can store only the same type of address. Void pointer can store any type of variable address. But before going to use void pointer, explicit type casting should be provided. Void pointer takes 2 bytes and used in dynamic memory allocation.

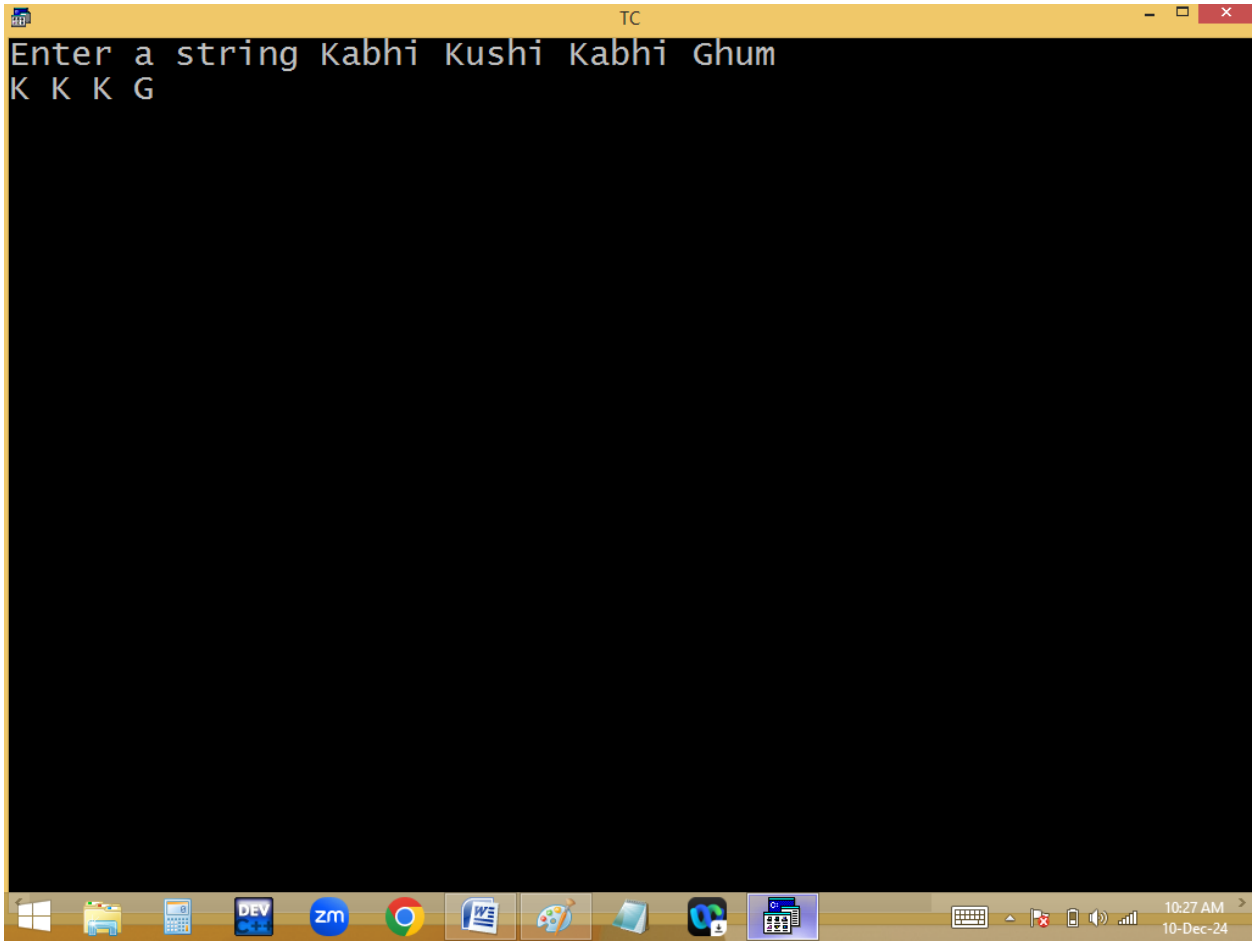

TC
File Edit Run Compile Project Options Debug
Line 13 Col 44 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10; float b=1.2; char c='X';
void *p;
clrscr();
p = &a;
printf("a=%d\n",*(int*)p); /* explicit type casting */
p = &b;
printf("b=%f\n",*(float*)p);
p = &c;
printf("c=%c\n",*(char*)p);
printf("void ptr size %d bytes",sizeof(p));
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:23 AM
10-Dec-24

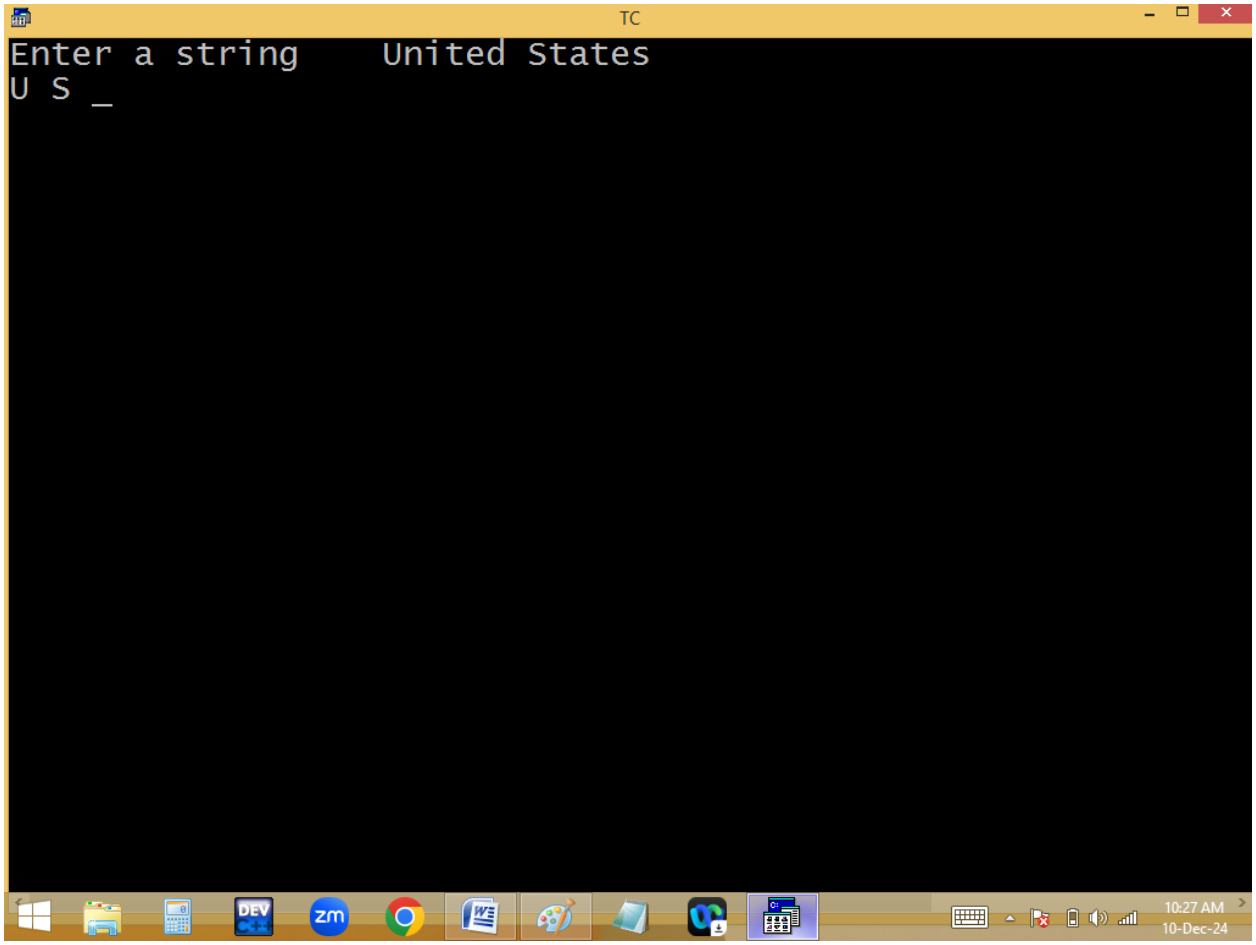


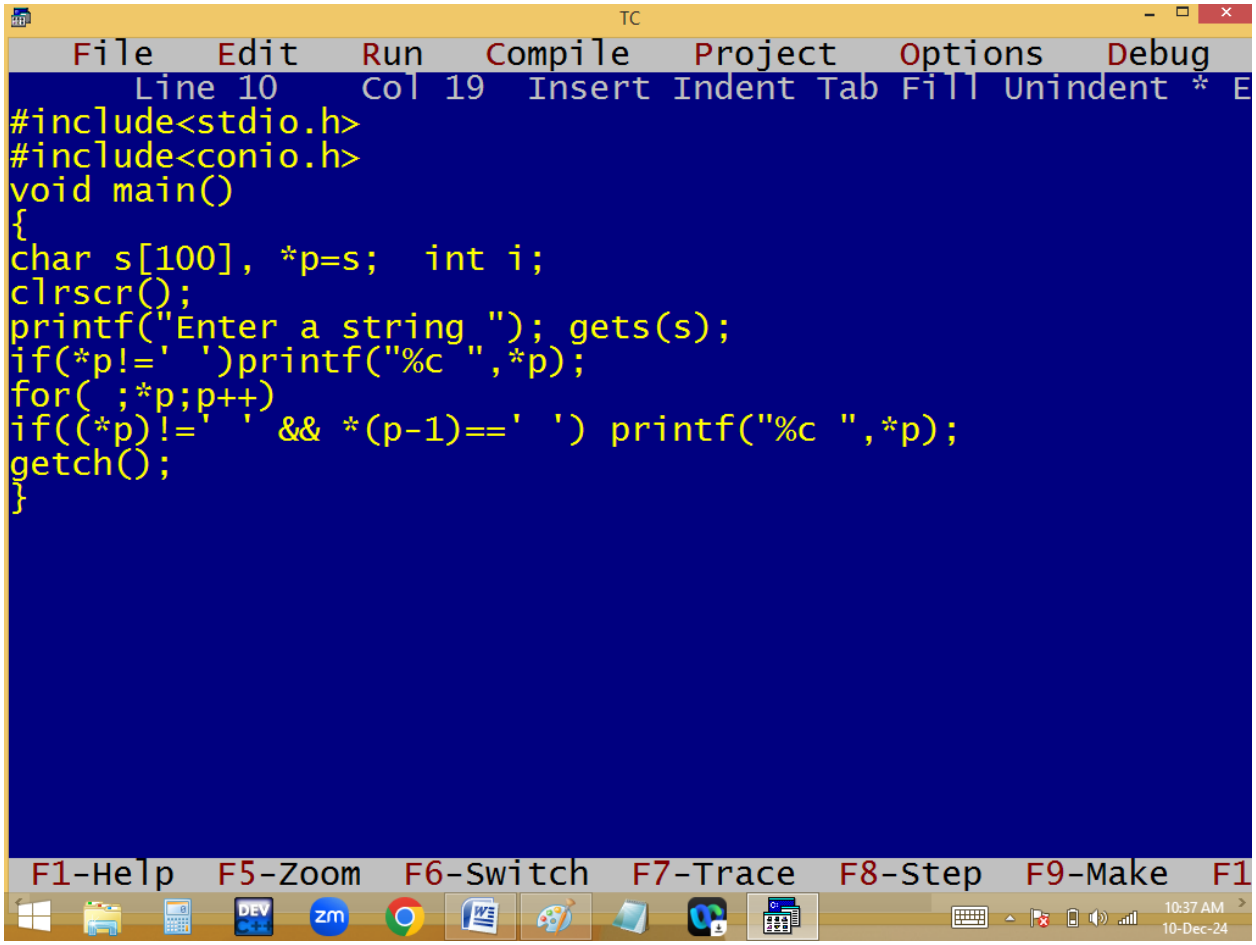
Abbreviation:

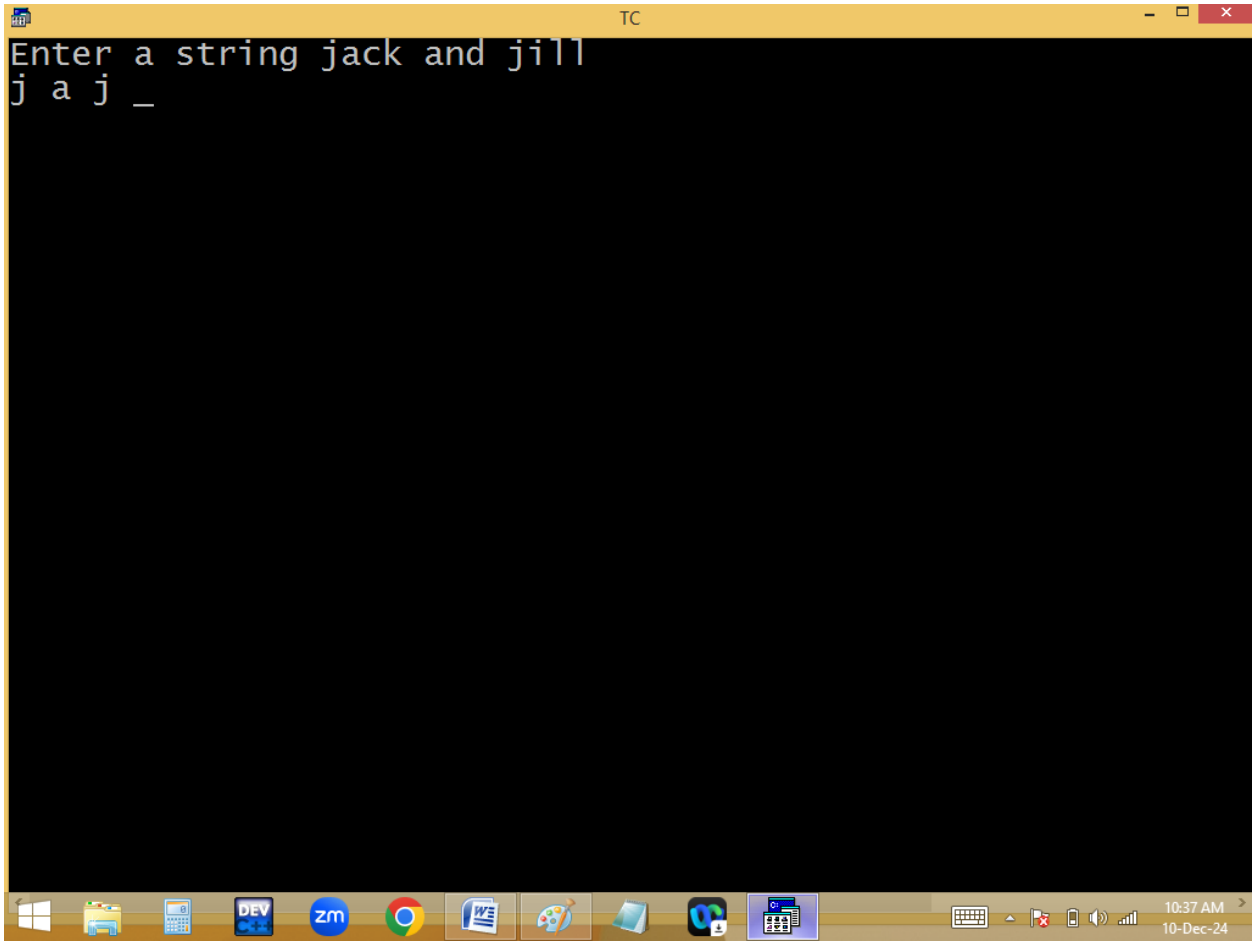
Ranam Rudhiram Raudram – R R R


TC
File Edit Run Compile Project Options Debug
Line 11 Col 1 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p; int i;
clrscr();
printf("Enter a string "); gets(s);
if(s[0]!=' ')printf("%c ",s[0]);
for(i=0;s[i]!='\0';i++)
if(s[i]!=' ' && s[i-1]==' ') printf("%c ",s[i]);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:27 AM
10-Dec-24


TC
Enter a string Kabhi Kushi Kabhi Ghum
K K K G
10:27 AM
10-Dec-24

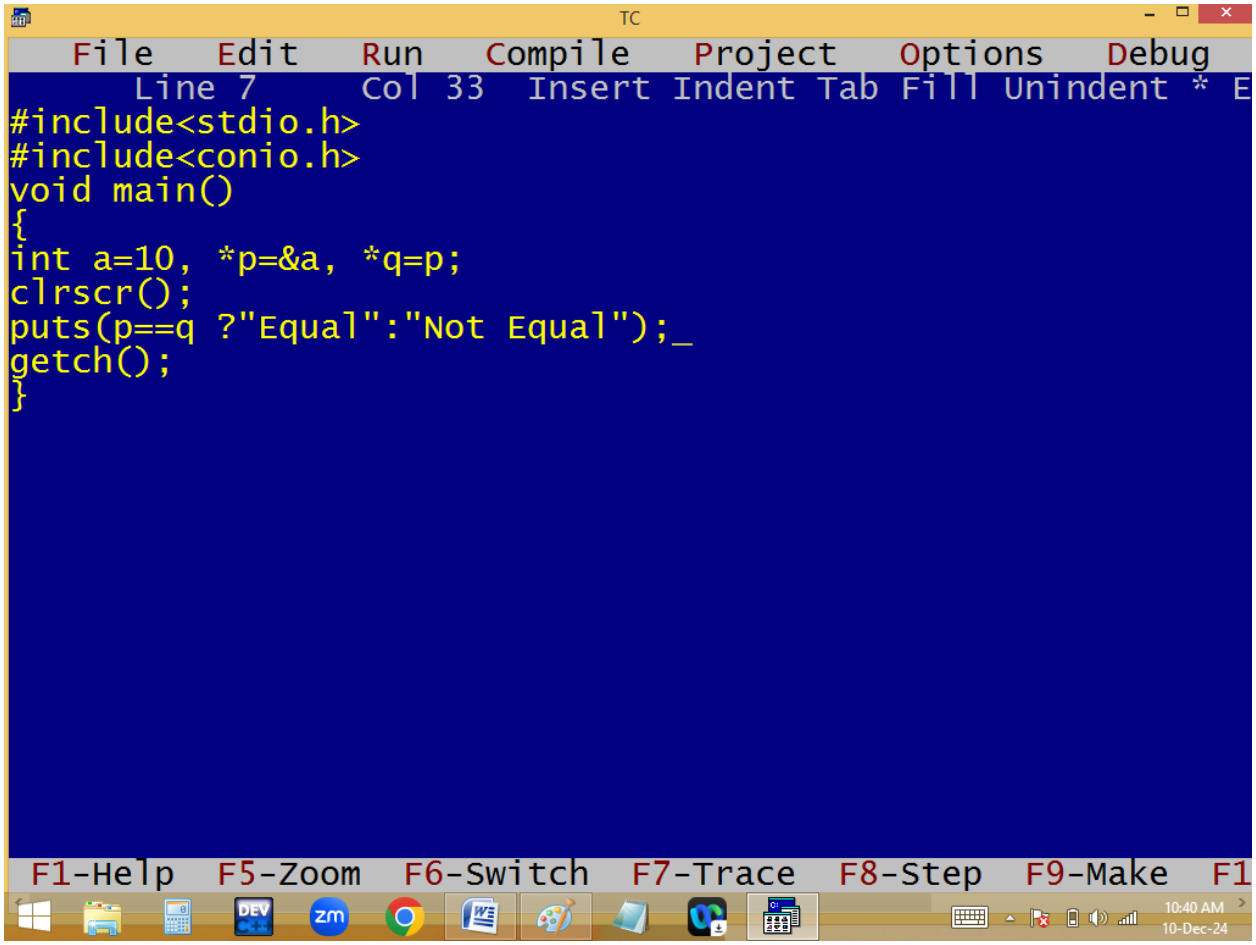

TC
Enter a string United States
U S _
DEV
zm
10:27 AM
10-Dec-24

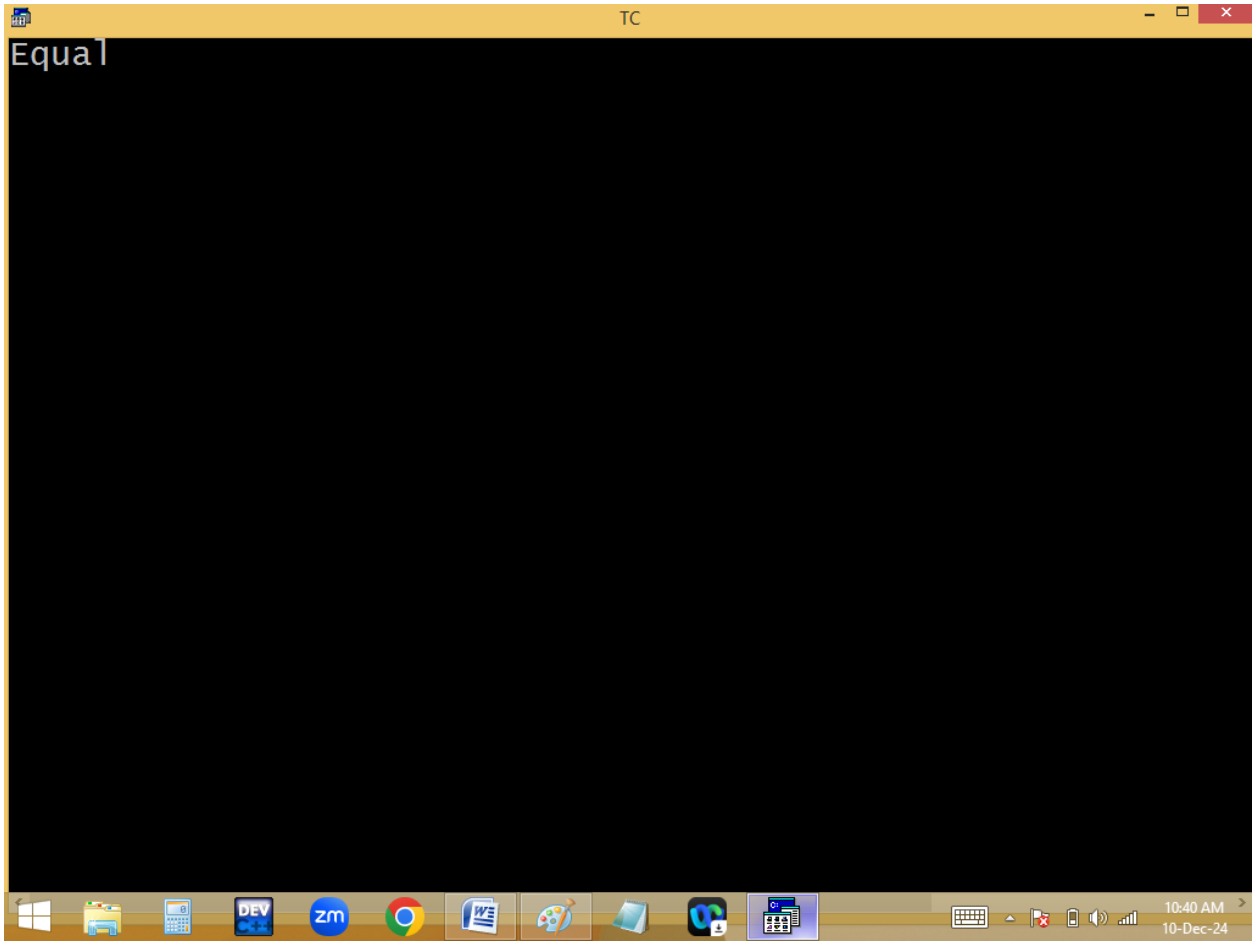

TC
File Edit Run Compile Project Options Debug
Line 10 Col 19 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
char s[100], *p=s; int i;
clrscr();
printf("Enter a string "); gets(s);
if(*p!=' ')printf("%c ",*p);
for(;*p;p++)
if((*p)!=' ' && *(p-1)==' ') printf("%c ",*p);
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:37 AM
10-Dec-24

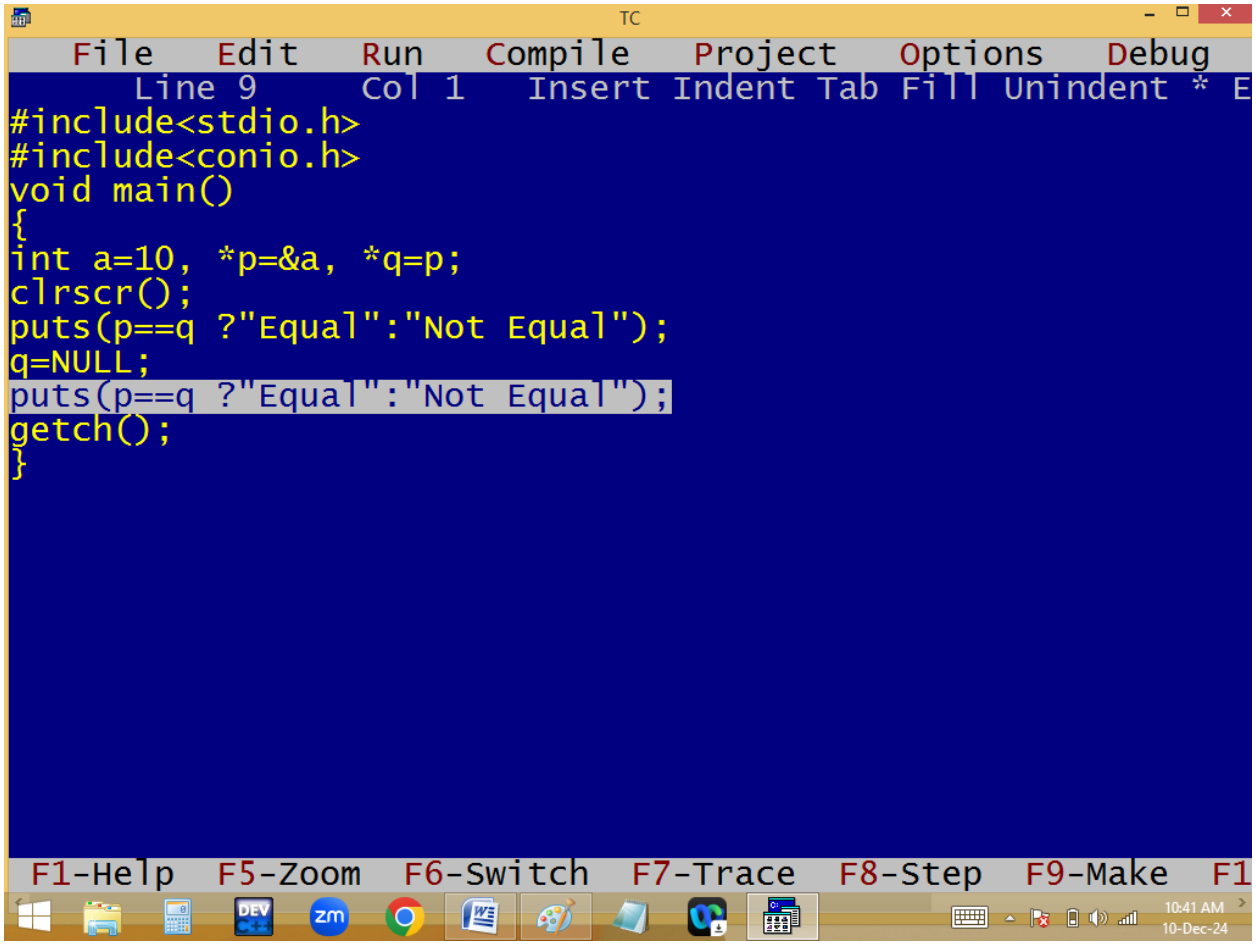


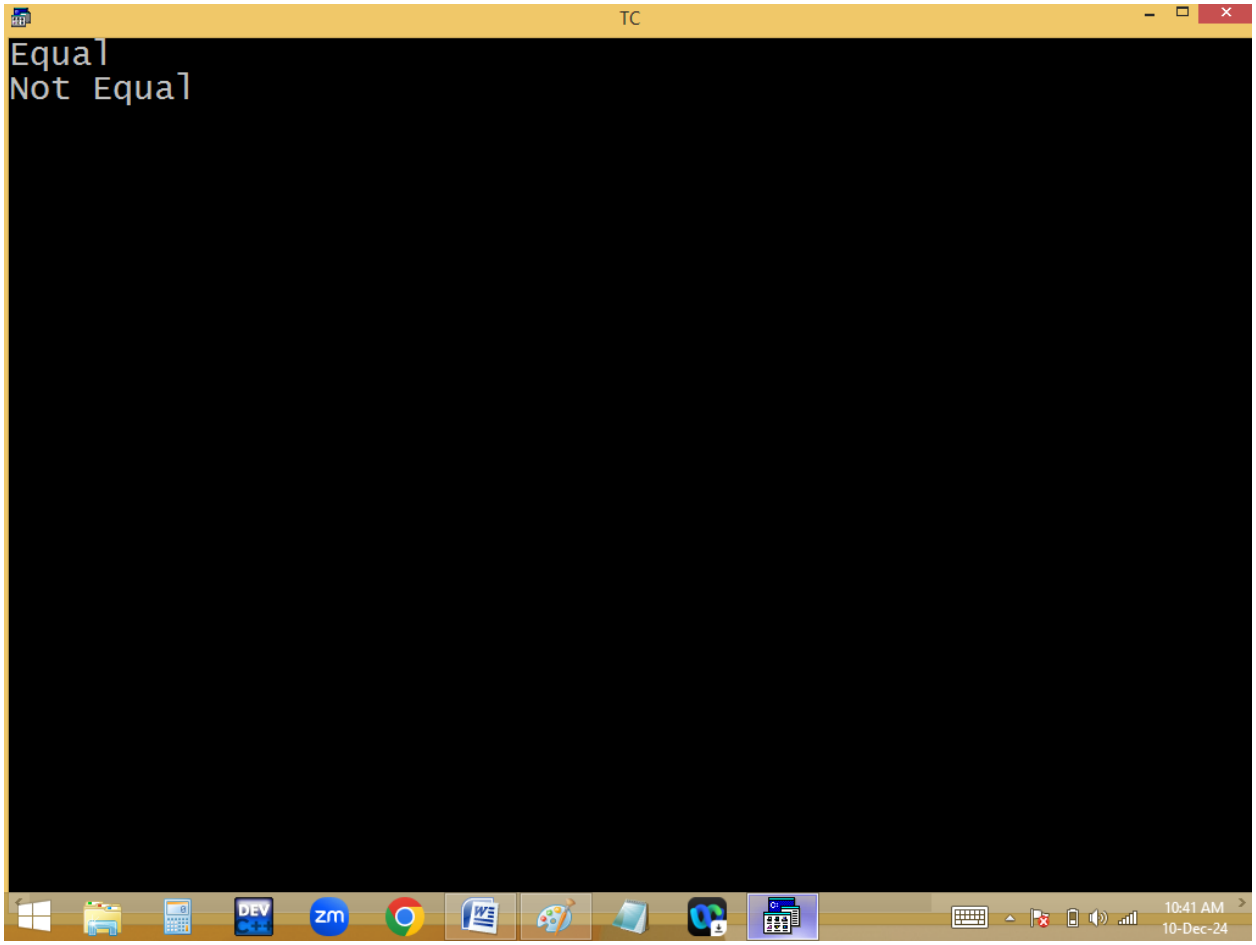
Pointer arithmetic:

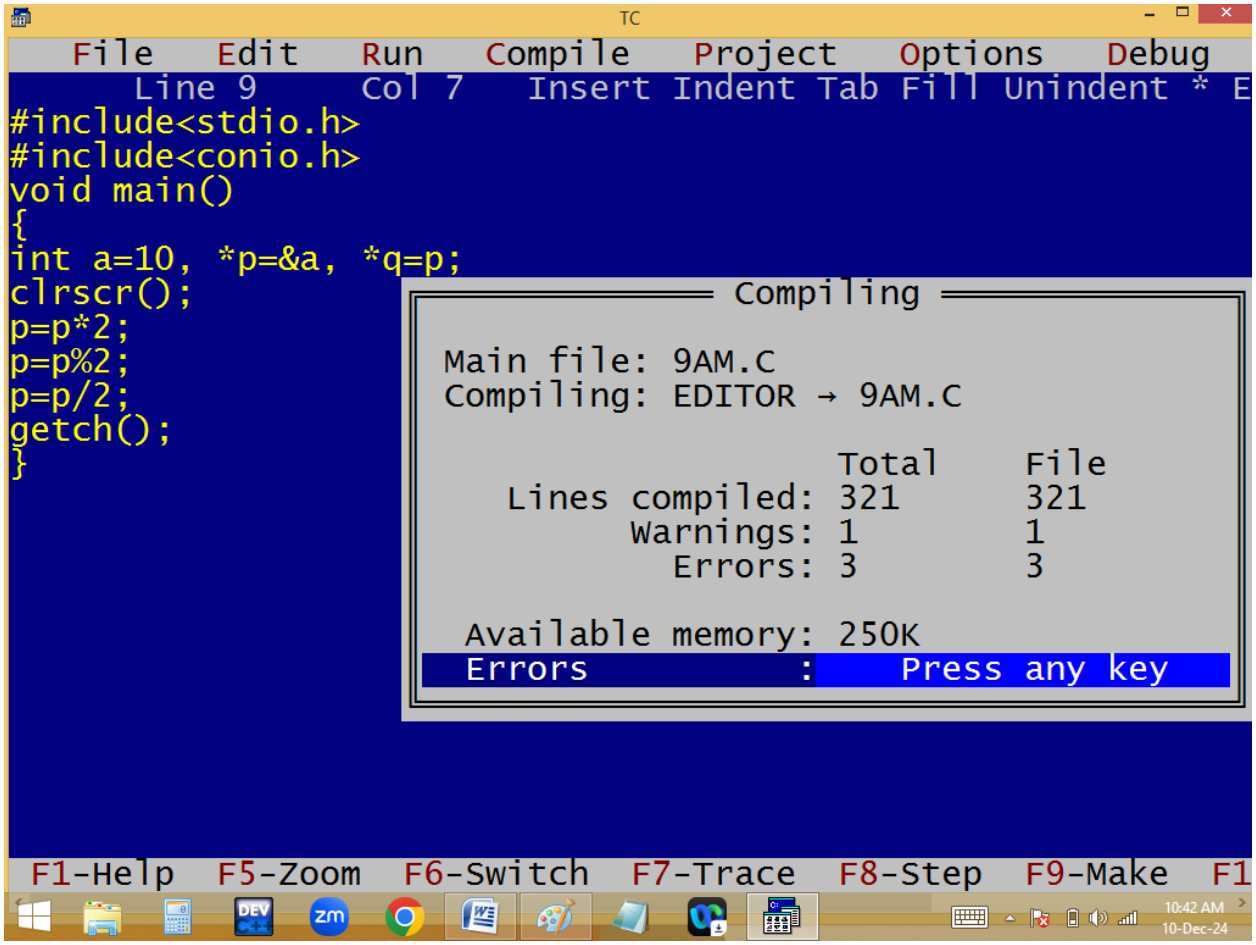
Like normal variables we can do copy, comparison, +, -, ++ and -- on pointers also. But we can't do *, % and / on pointers.

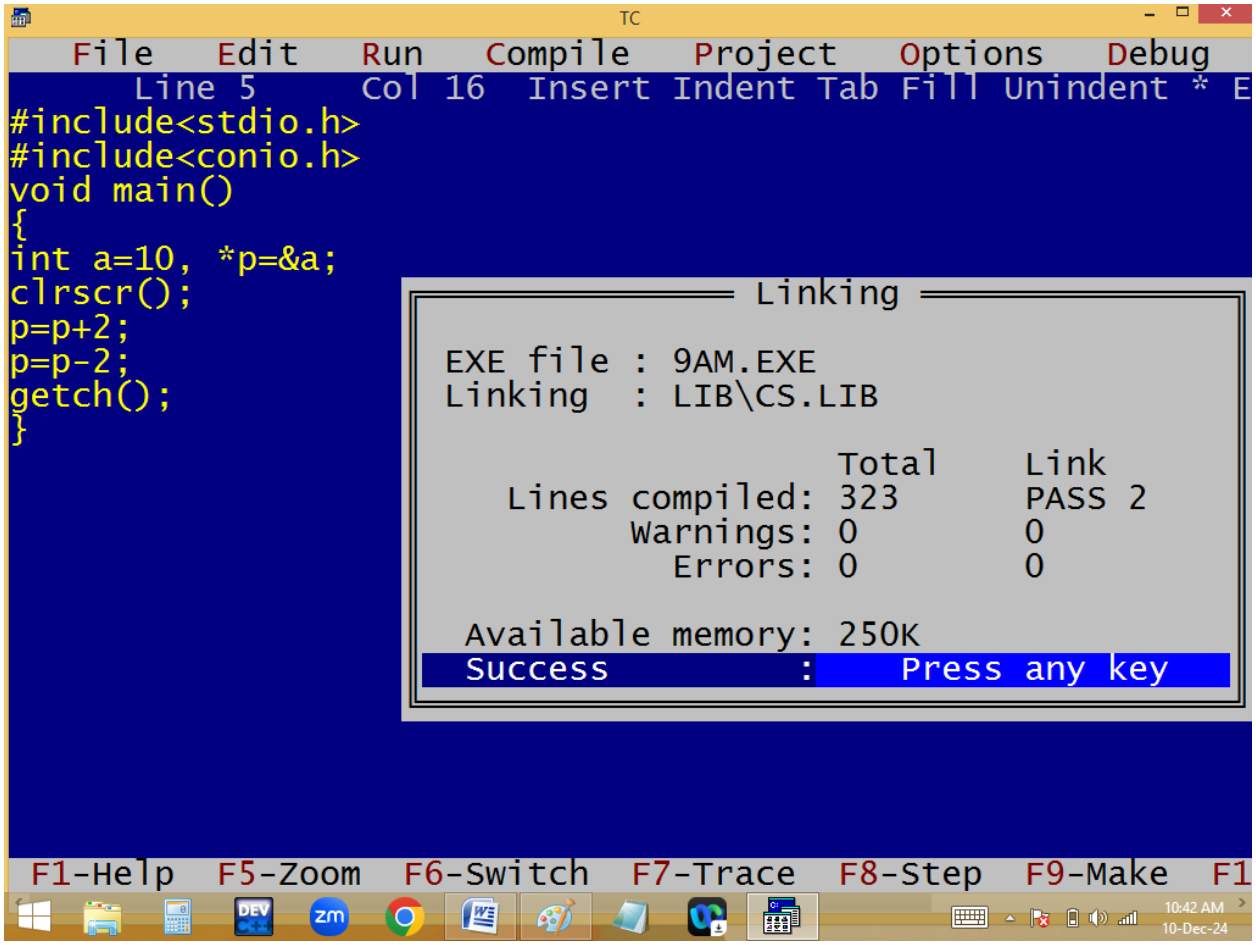

TC
File Edit Run Compile Project Options Debug
Line 7 Col 33 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a, *q=p;
clrscr();
puts(p==q ?"Equal":"Not Equal");_
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:40 AM
10-Dec-24

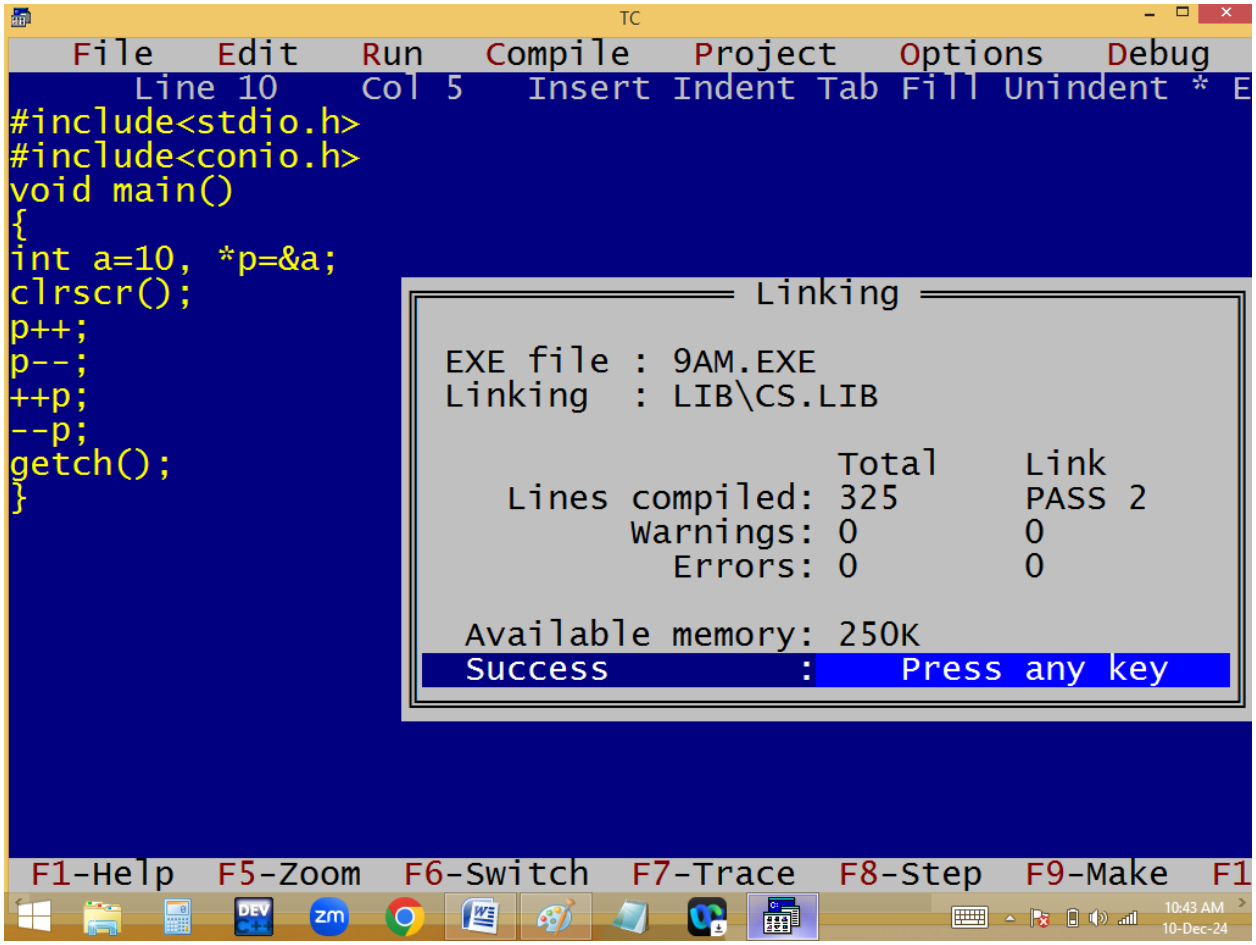

TC
Equal
10:40 AM
10-Dec-24


TC
File Edit Run Compile Project Options Debug
Line 9 Col 1 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a, *q=p;
clrscr();
puts(p==q ?"Equal":"Not Equal");
q=NULL;
puts(p==q ?"Equal":"Not Equal");
getch();
}
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:41 AM
10-Dec-24


TC
Equal
Not Equal
10:41 AM
10-Dec-24


TC
File Edit Run Compile Project Options Debug
Line 9 Col 7 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a, *q=p;
clrscr();
p=p*2;
p=p%2;
p=p/2;
getch();
}
Compiling
Main file: 9AM.C
Compiling: EDITOR → 9AM.C
Total File
Lines compiled: 321 321
Warnings: 1 1
Errors: 3 3
Available memory: 250K
Errors : Press any key
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:42 AM
10-Dec-24


TC
File Edit Run Compile Project Options Debug
Line 5 Col 16 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a;
clrscr();
p=p+2;
p=p-2;
getch();
}
Linking
EXE file : 9AM.EXE
Linking : LIB\CS.LIB
Total Link
Lines compiled: 323 PASS 2
Warnings: 0 0
Errors: 0 0
Available memory: 250K
Success : Press any key
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:42 AM
10-Dec-24


TC
File Edit Run Compile Project Options Debug
Line 10 Col 5 Insert Indent Tab Fill Unindent * E
#include<stdio.h>
#include<conio.h>
void main()
{
int a=10, *p=&a;
clrscr();
p++;
p--;
++p;
--p;
getch();
}
Linking
EXE file : 9AM.EXE
Linking : LIB\CS.LIB
Total Link
Lines compiled: 325 PASS 2
Warnings: 0 0
Errors: 0 0
Available memory: 250K
Success : Press any key
F1-Help F5-Zoom F6-Switch F7-Trace F8-Step F9-Make F1
10:43 AM
10-Dec-24