Ternary / Conditional operator(? :)

It is similar to if else / ladder if in working style.

It allows to complete if else / ladder if in a single statement.

When we are working with if else/ladder if it is going to take more than one line of statements. Ternary operator is going to finish the same task in a single statement.

But the difference between if …else and ternary operator is ternary operator supports only one statement at a time and if supports any number of statements.

It is having 3 expressions. Hence it is called ternary operator.

It is starting with a condition. Hence it is called conditional operator.

Syntax:

condition ? true statement : false statement ;

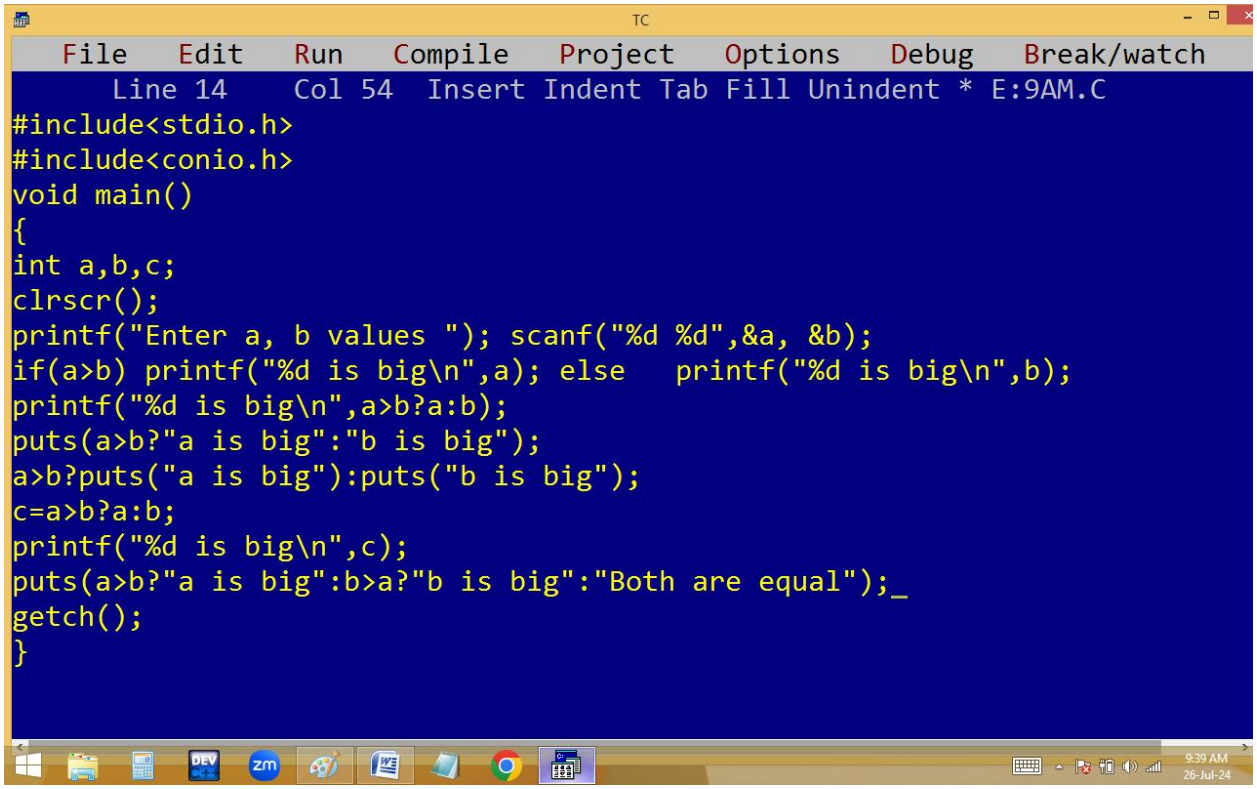
exp1/op1 exp2/op2 exp3/op3

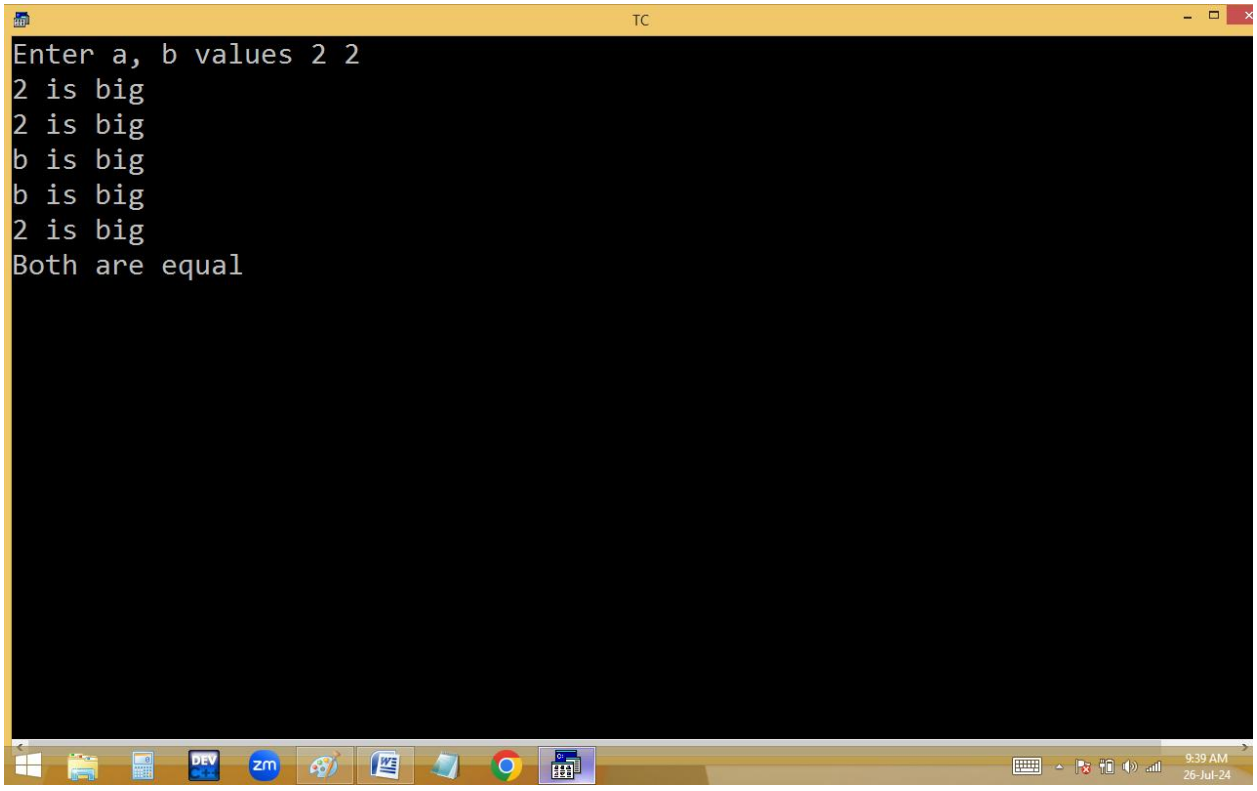
If condition true, statement after **?** executed.

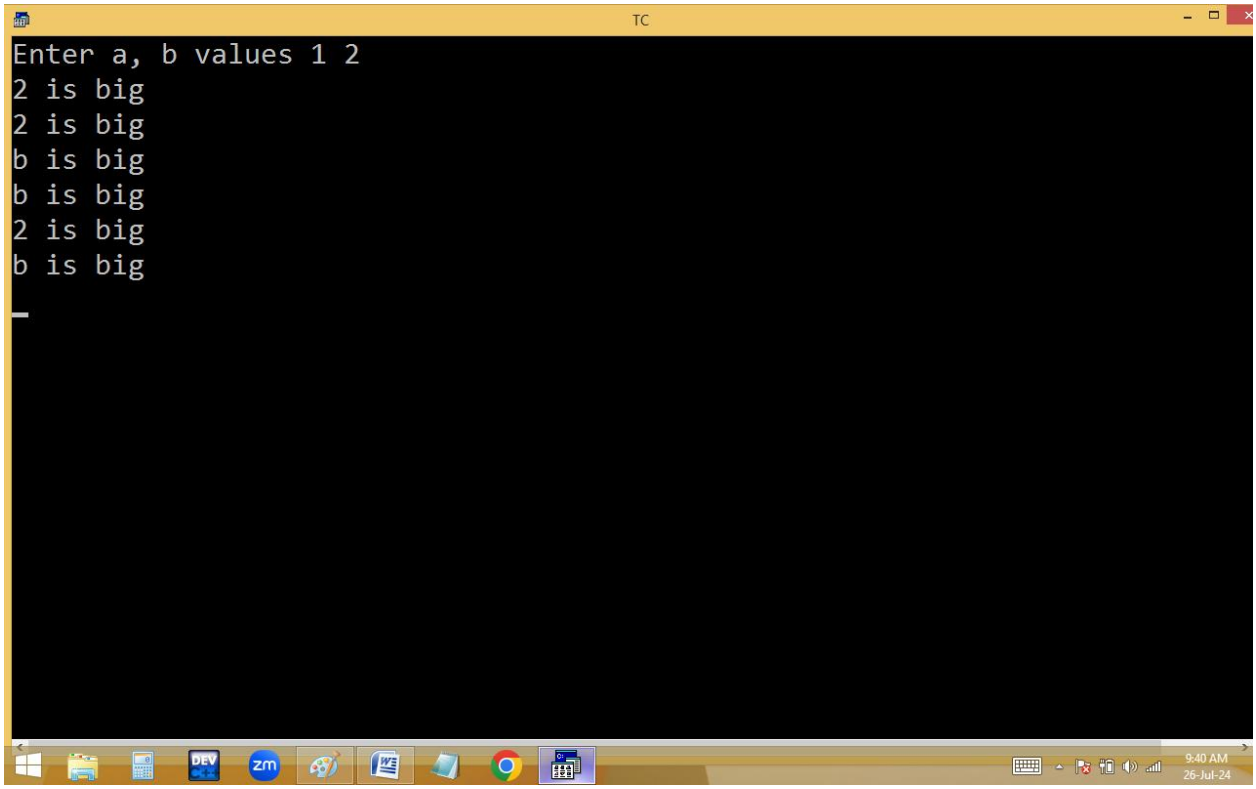
If condition false, statement after **:** is executed.

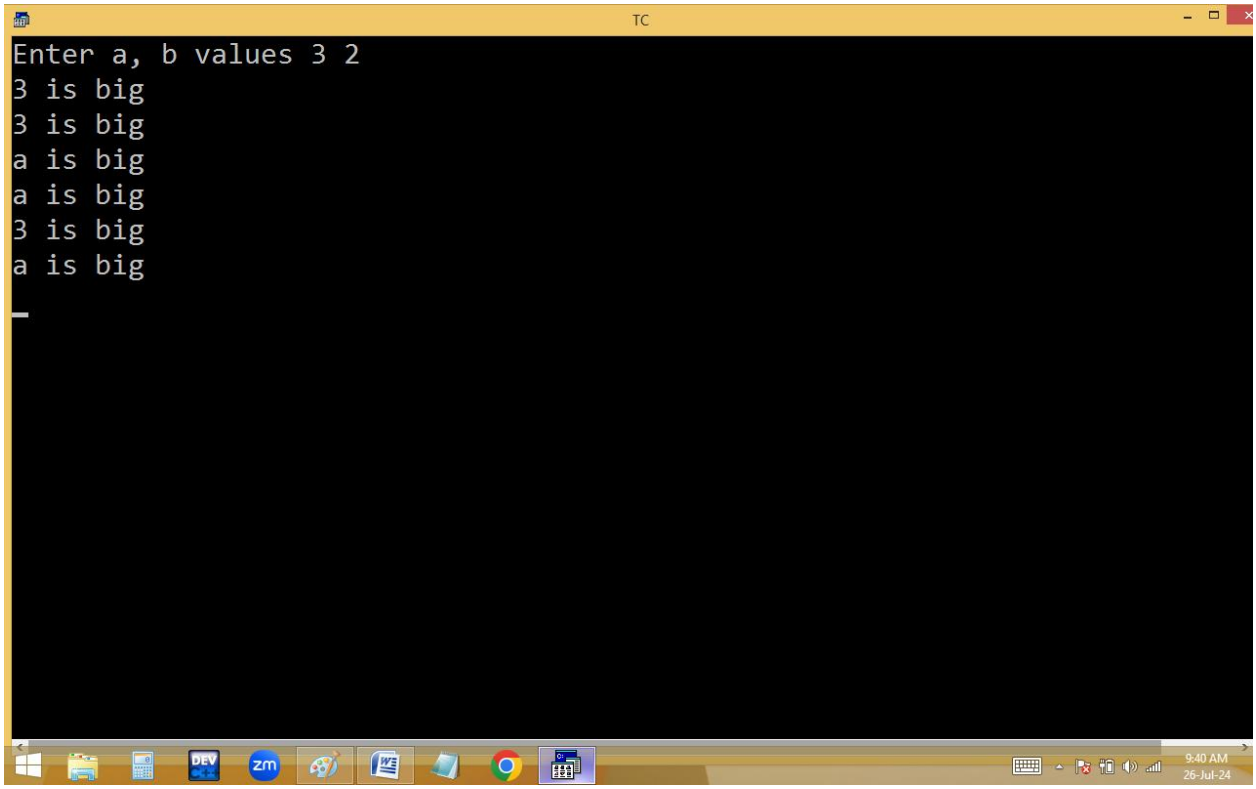
When compared with if else, conditional operator **performance is high**.

Eg:Finding big in two no's using ternary operator.

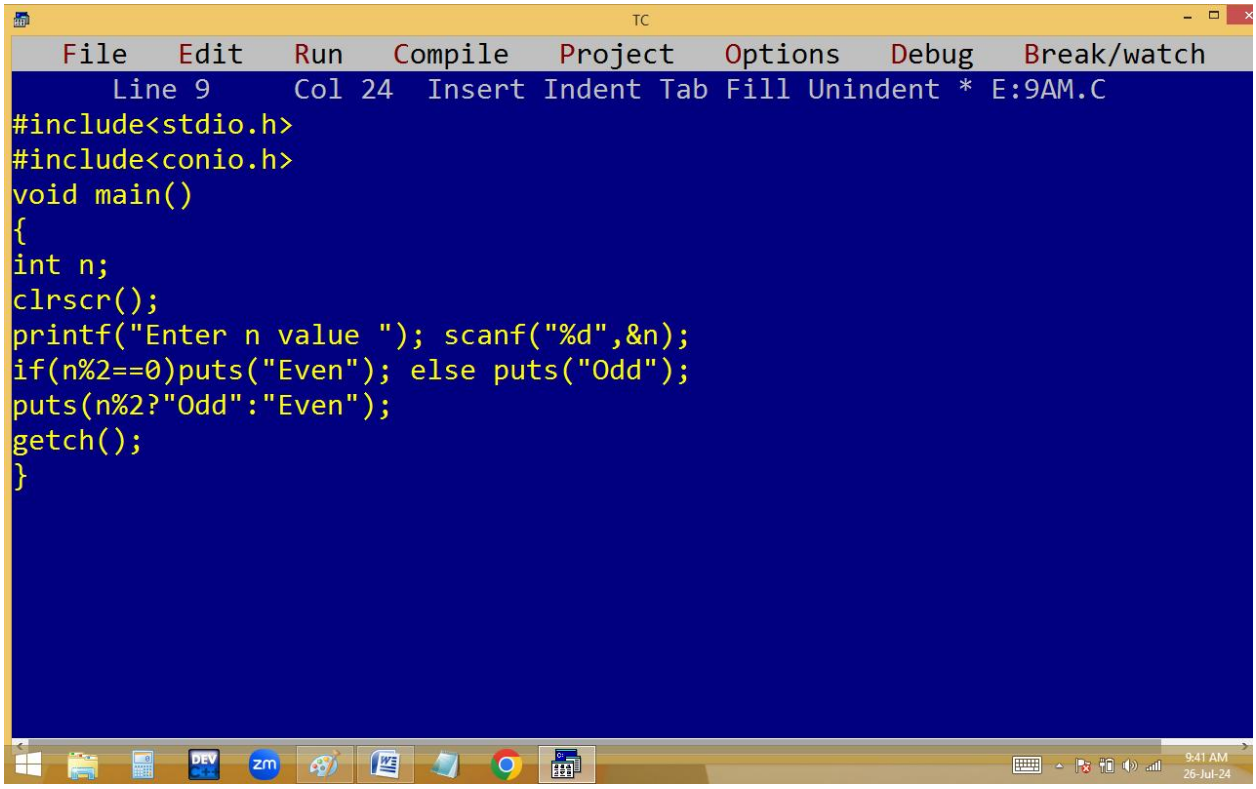

TC
File Edit Run Compile Project Options Debug Break/watch
Line 14 Col 54 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a,b,c;
clrscr();
printf("Enter a, b values "); scanf("%d %d",&a, &b);
if(a>b) printf("%d is big\n",a); else printf("%d is big\n",b);
printf("%d is big\n",a>b?a:b);
puts(a>b?"a is big":"b is big");
a>b?puts("a is big"):puts("b is big");
c=a>b?a:b;
printf("%d is big\n",c);
puts(a>b?"a is big":b>a?"b is big":"Both are equal");
getch();
}

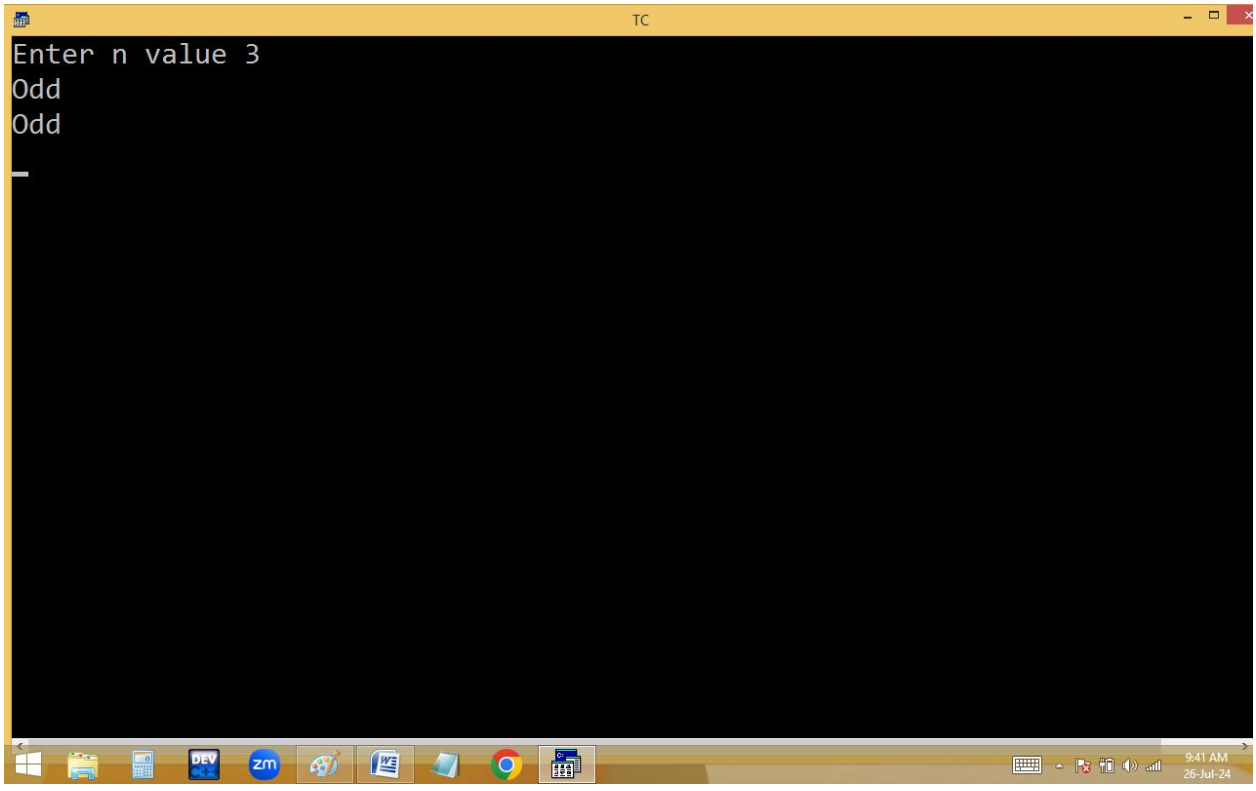

TC
Enter a, b values 2 2
2 is big
2 is big
b is big
b is big
2 is big
Both are equal

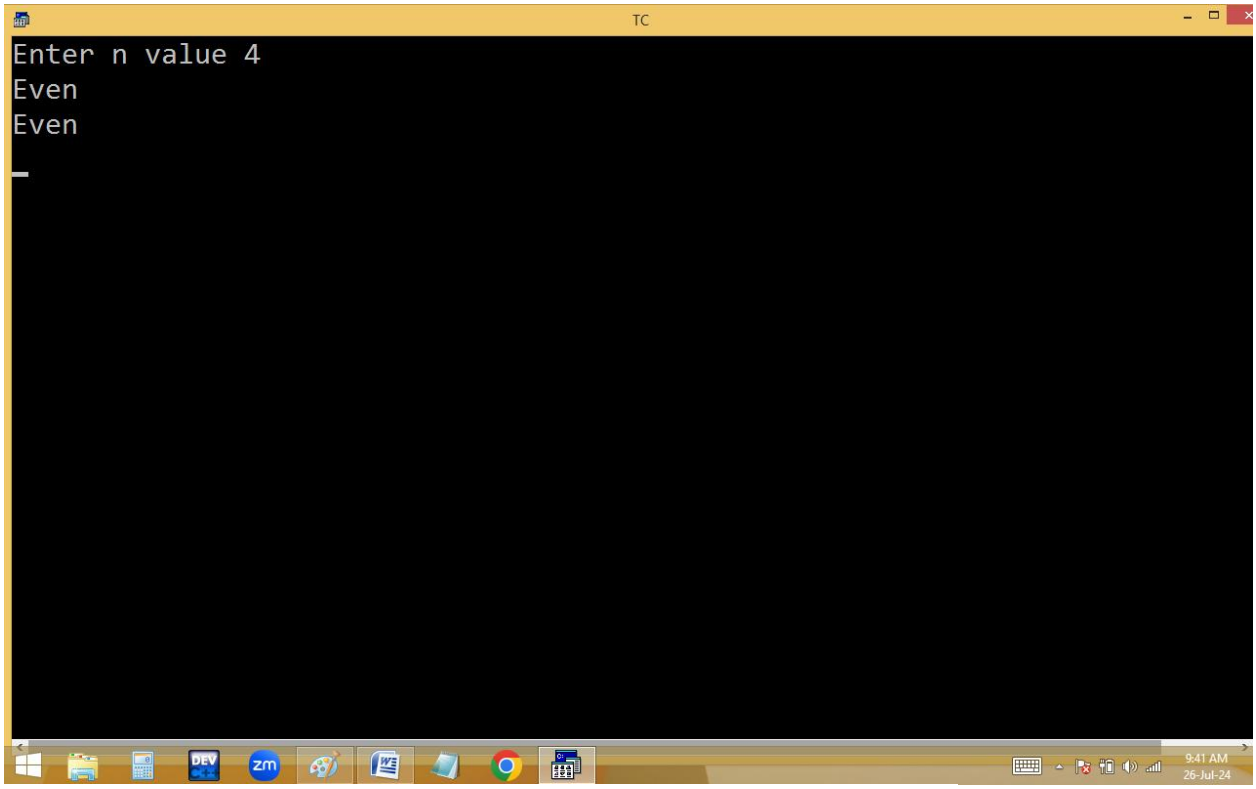

TC
Enter a, b values 1 2
2 is big
2 is big
b is big
b is big
2 is big
b is big

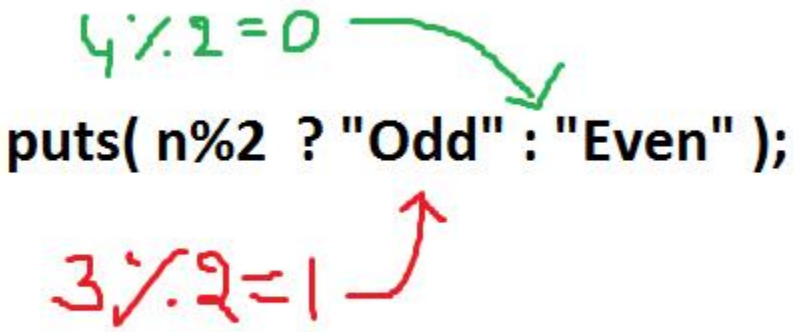

TC
Enter a, b values 3 2
3 is big
3 is big
a is big
a is big
3 is big
a is big

Finding even/odd using ternary operator:

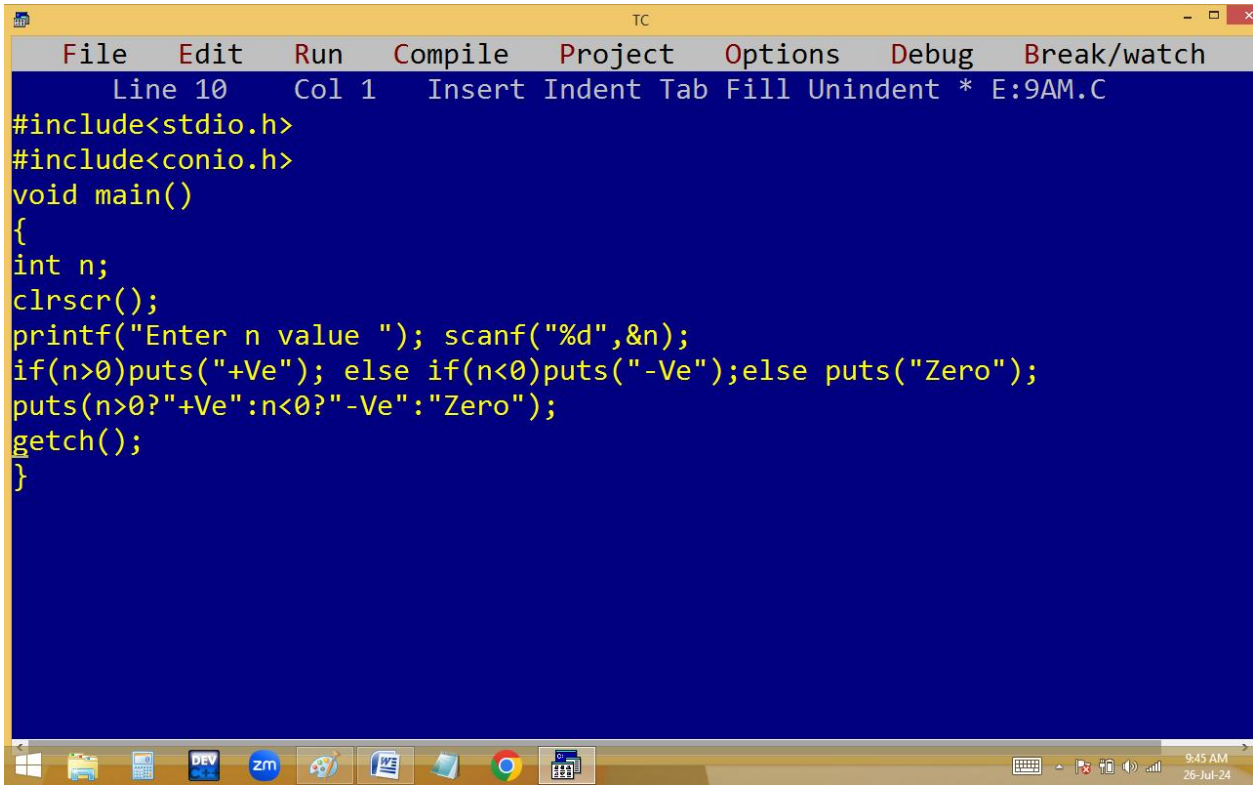


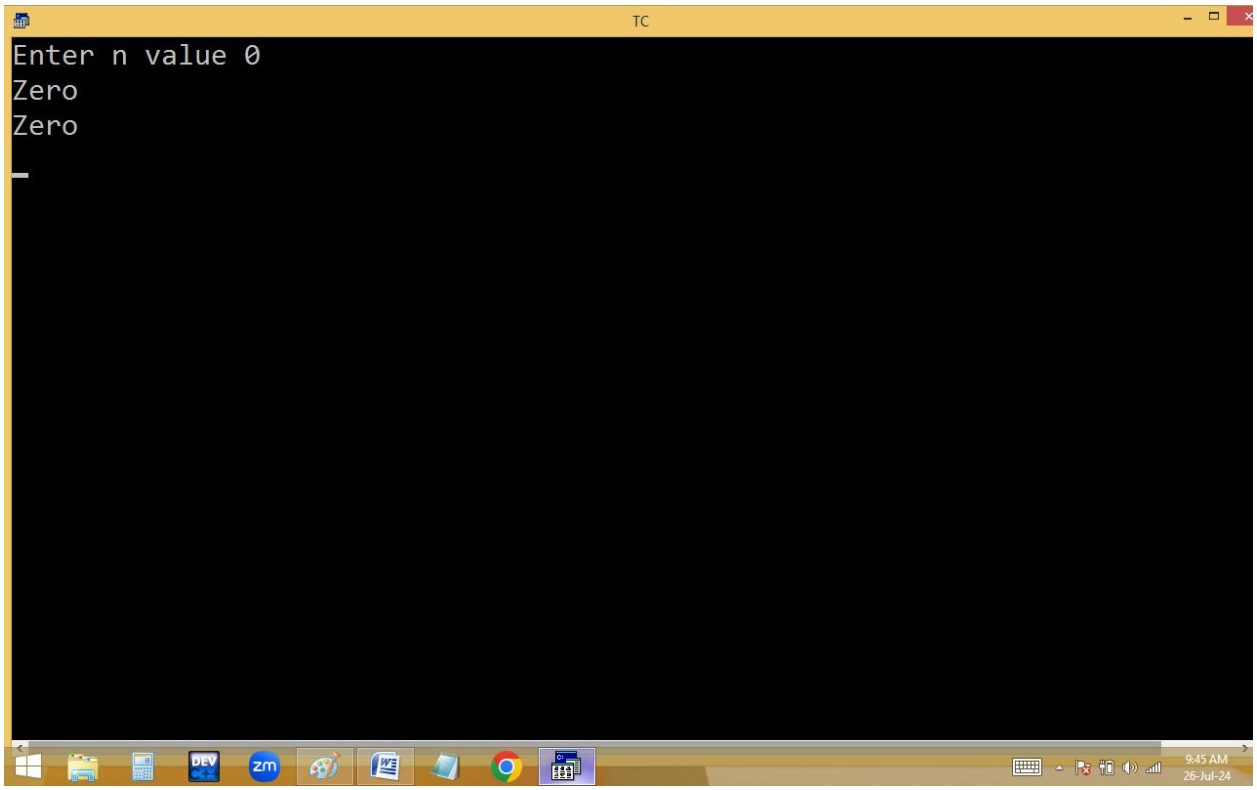


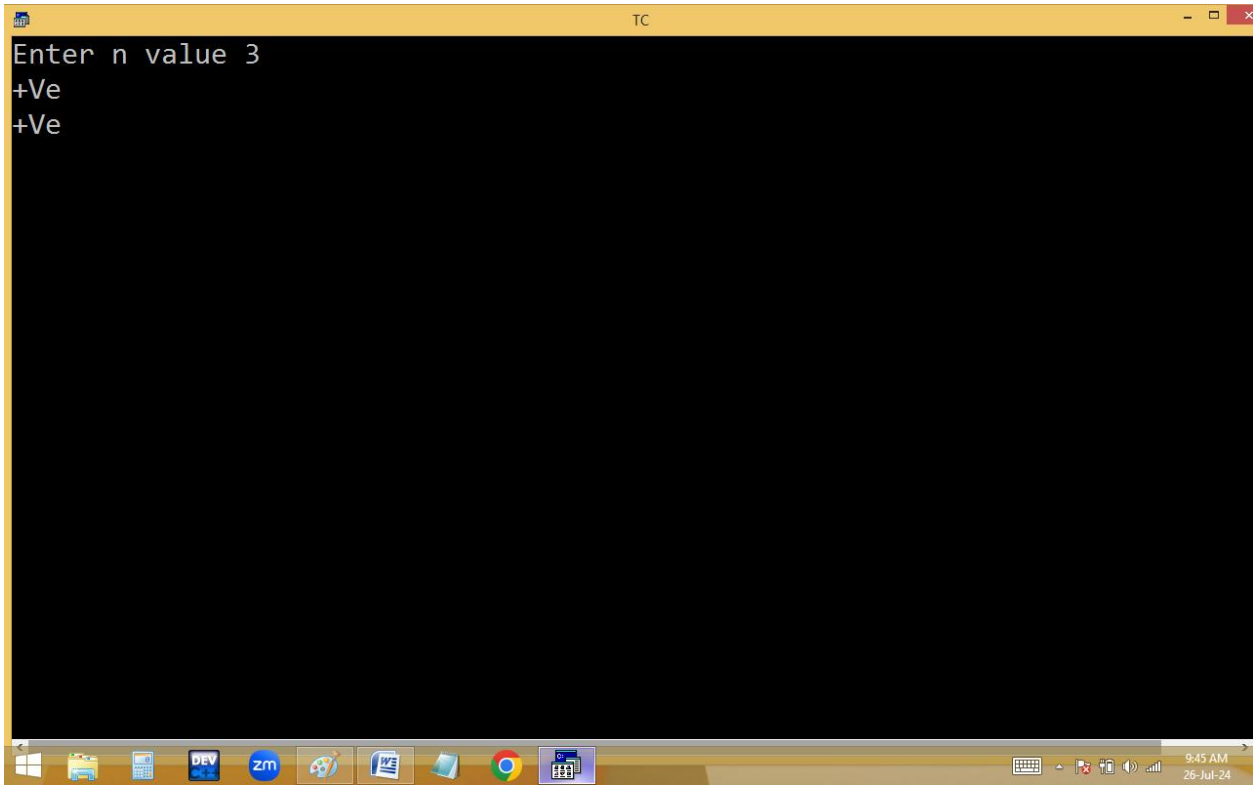


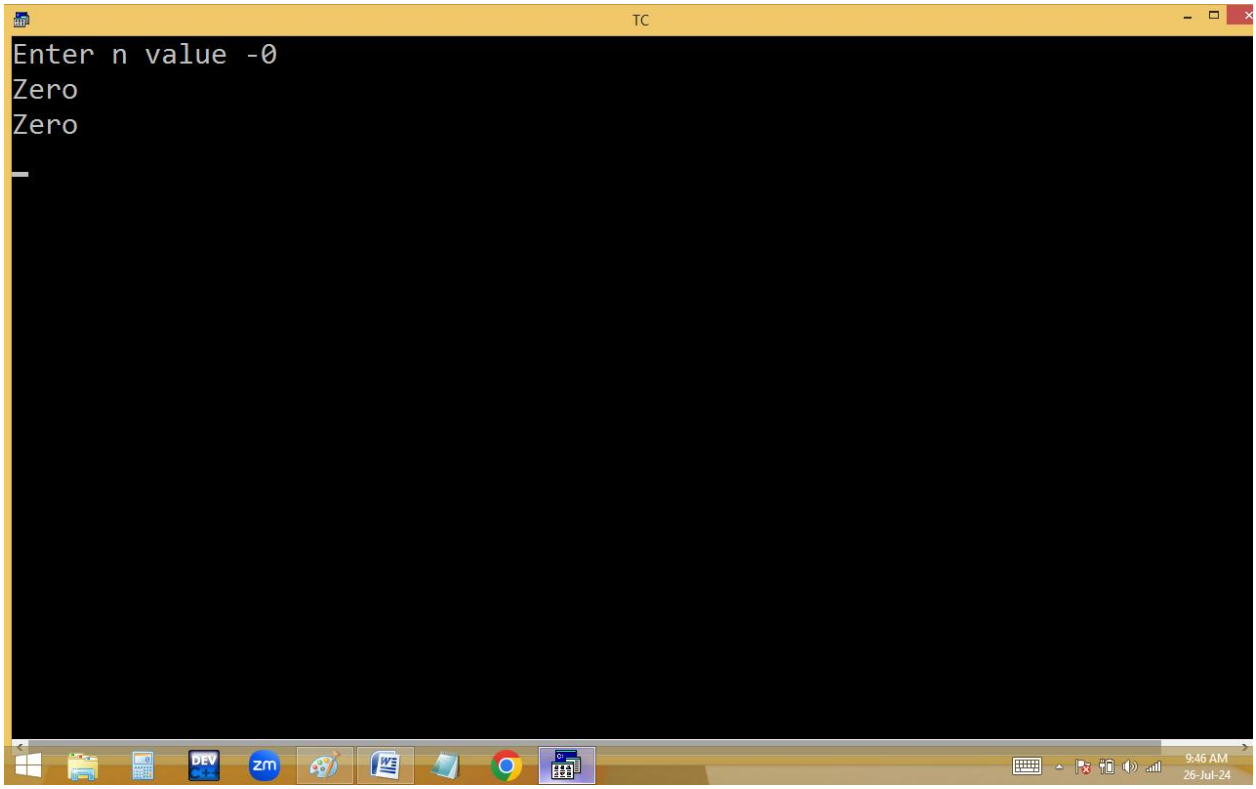


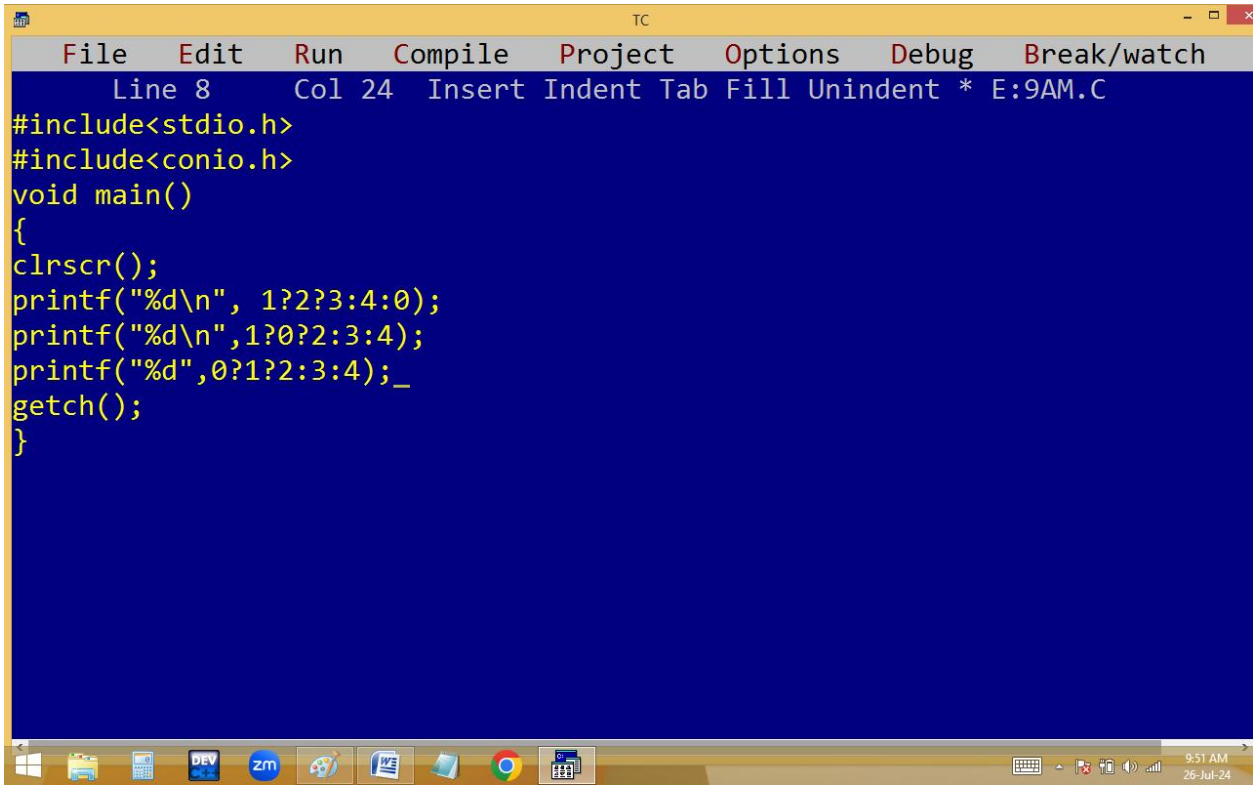
Finding +Ve / -Ve / 0 using ternary operator:

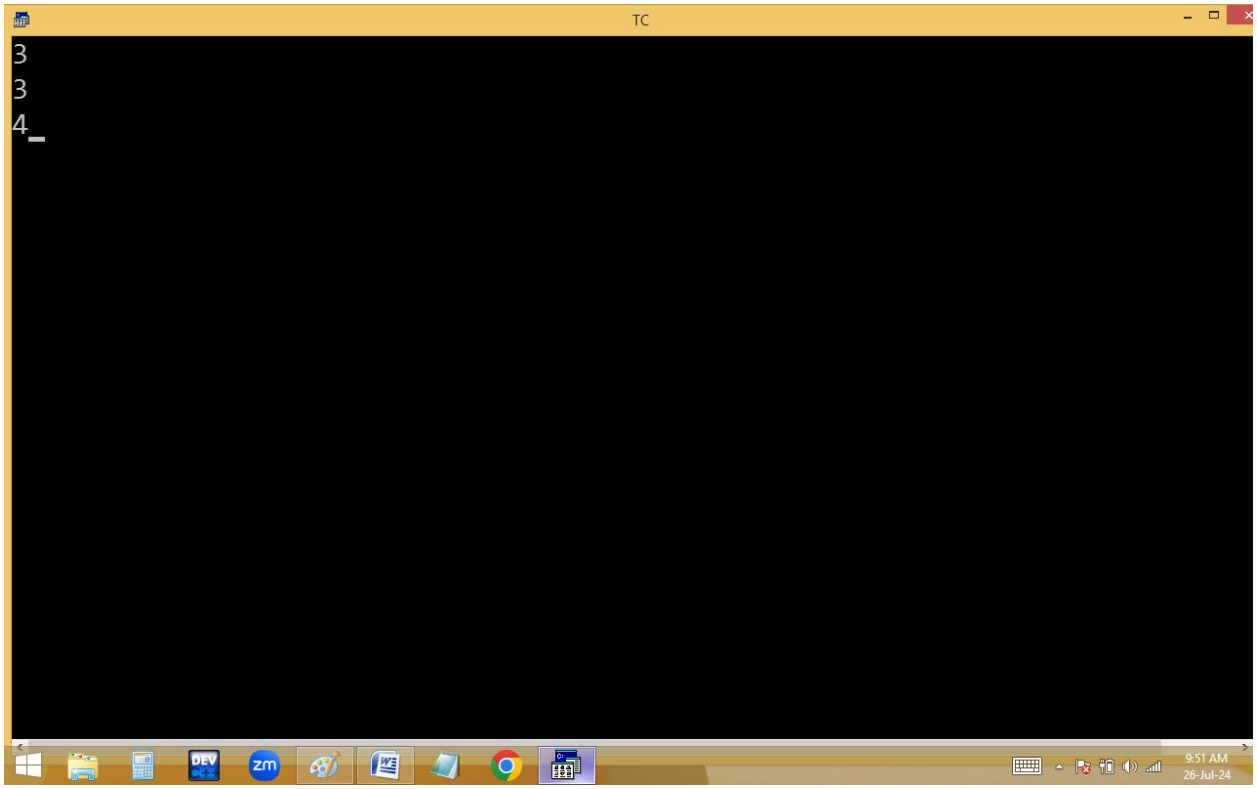

TC
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 1 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n;
clrscr();
printf("Enter n value "); scanf("%d",&n);
if(n>0)puts("+Ve"); else if(n<0)puts("-Ve");else puts("Zero");
puts(n>0?"+Ve":n<0?"-Ve":"Zero");
getch();
}

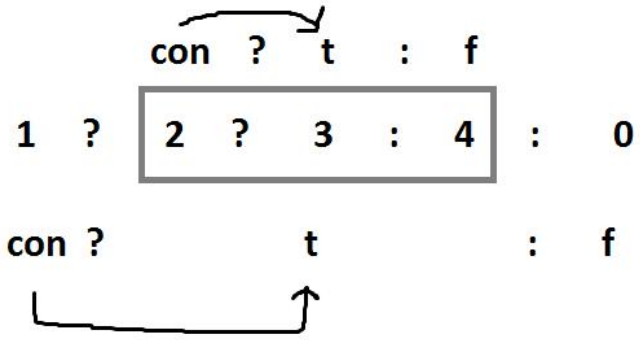

TC
Enter n value 0
Zero
Zero


TC
Enter n value 3
+Ve
+Ve
9:45 AM
26-Jul-24

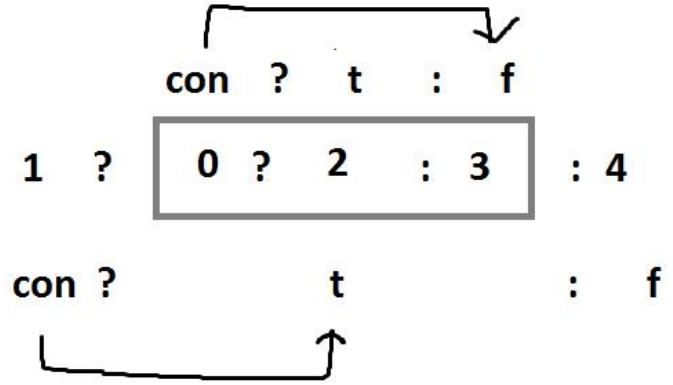

TC
Enter n value -0
Zero
Zero
9:46 AM
26-Jul-24


TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 24 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
printf("%d\n", 1?2?3:4:0);
printf("%d\n",1?0?2:3:4);
printf("%d",0?1?2:3:4);
getch();
}


TC
3
3
4

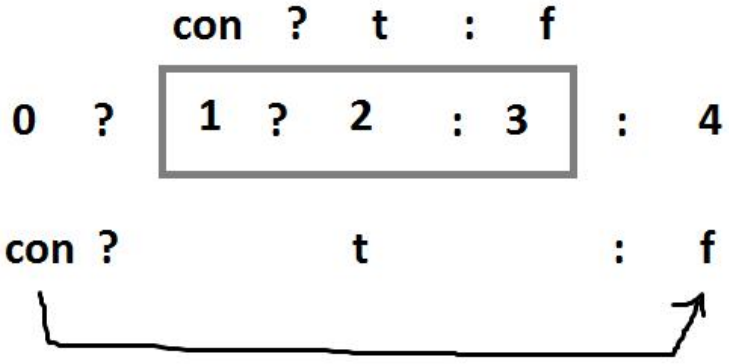



```
if( 1 )
{
if( 2 ) p( 3 );
else  p( 4 );
}
else  p( 0 )
```



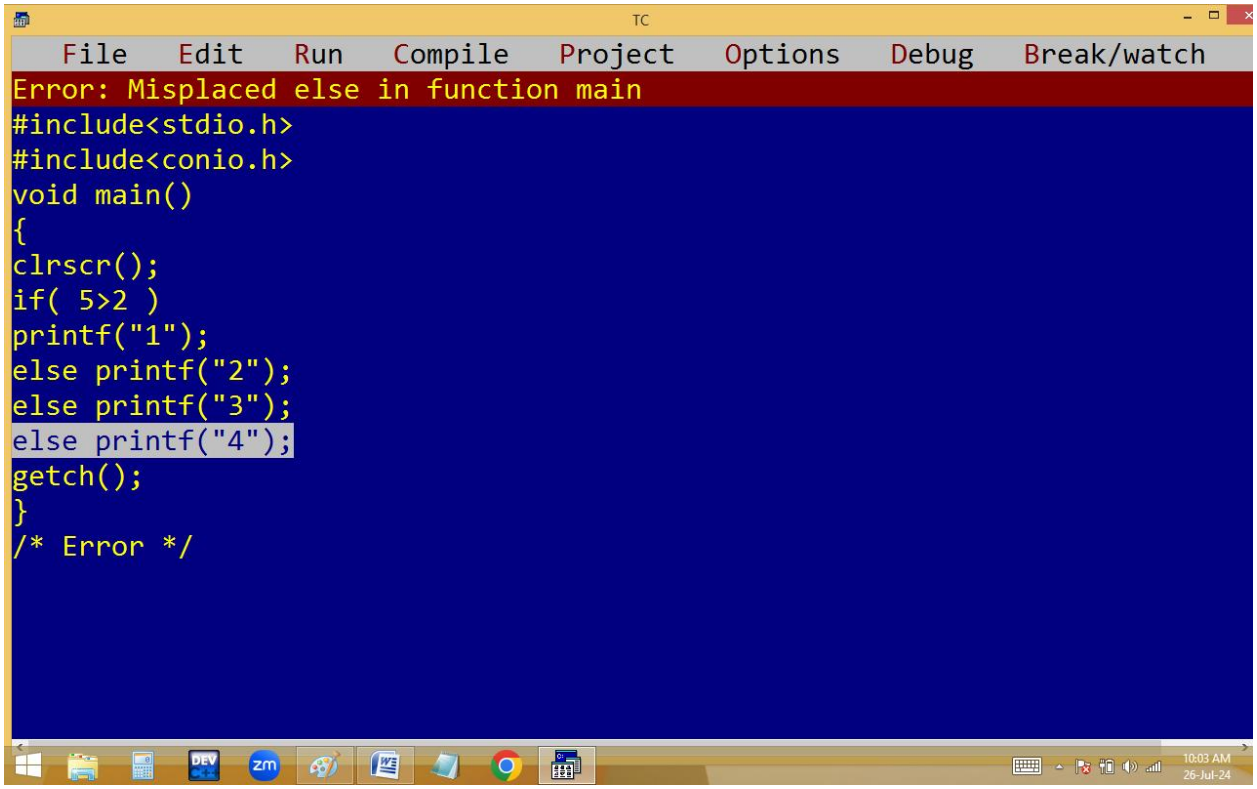
F

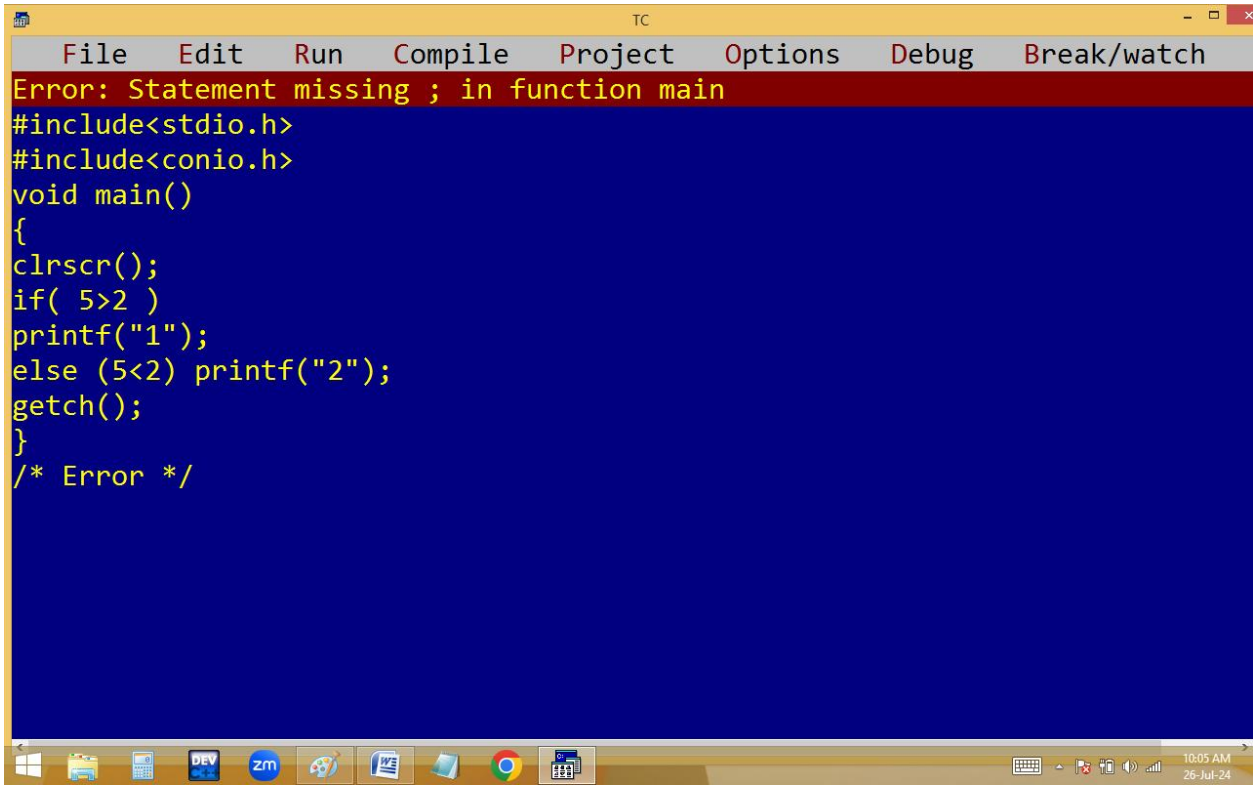
```
if( 1 )
{
if( 0 ) p( 2 );
else  p( 3 );
}
else  p( 4 )
```

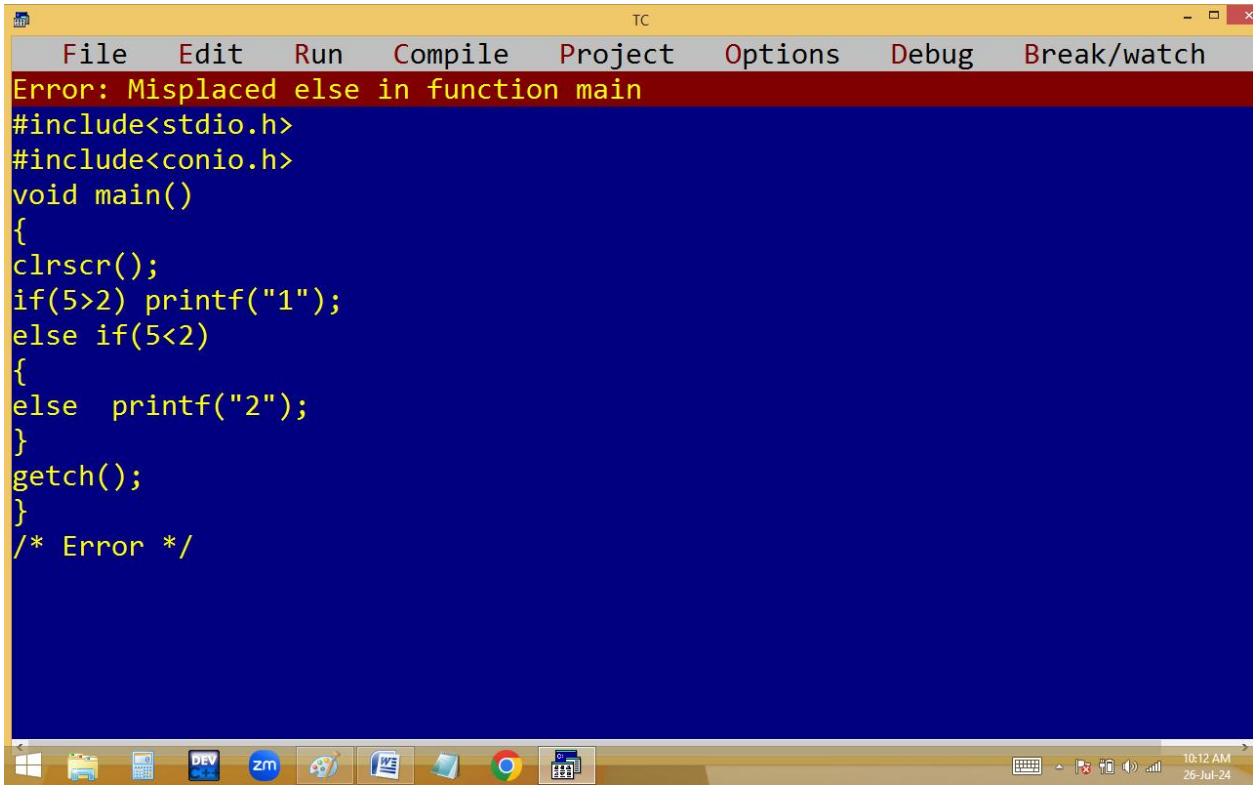


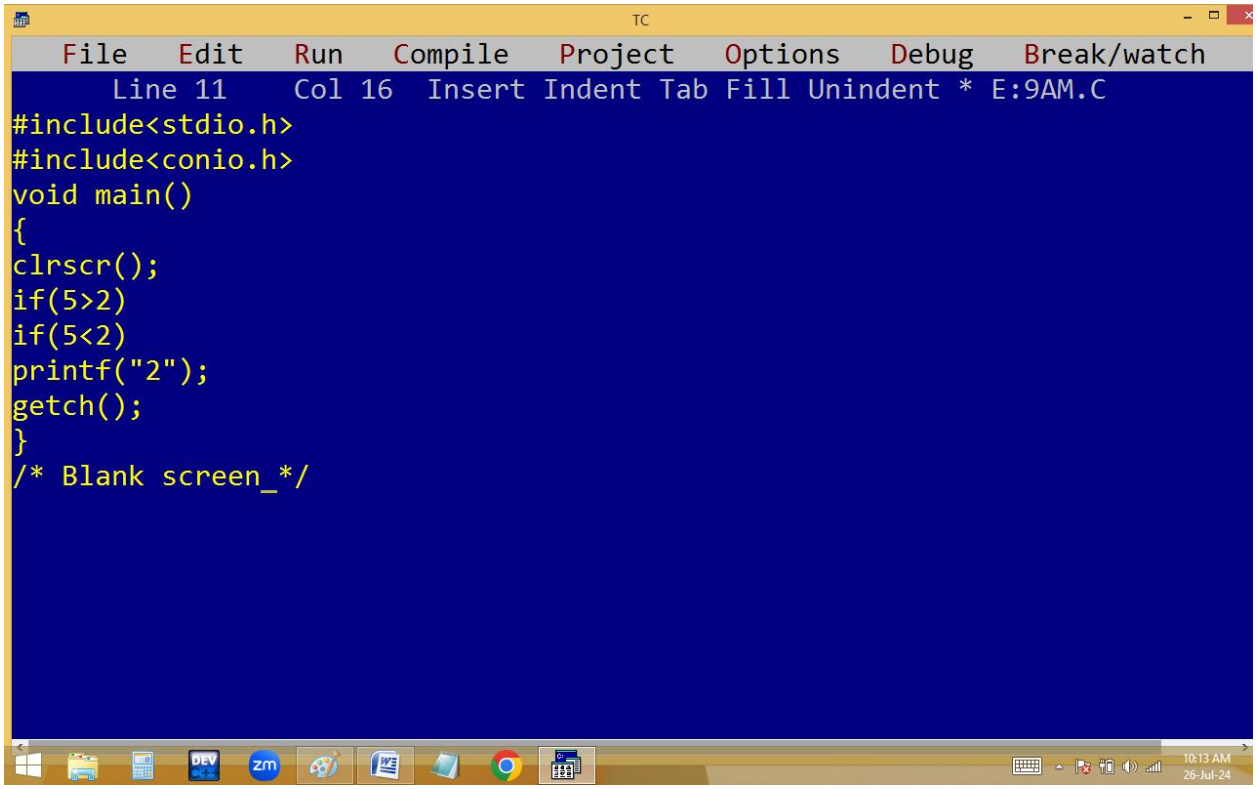
F

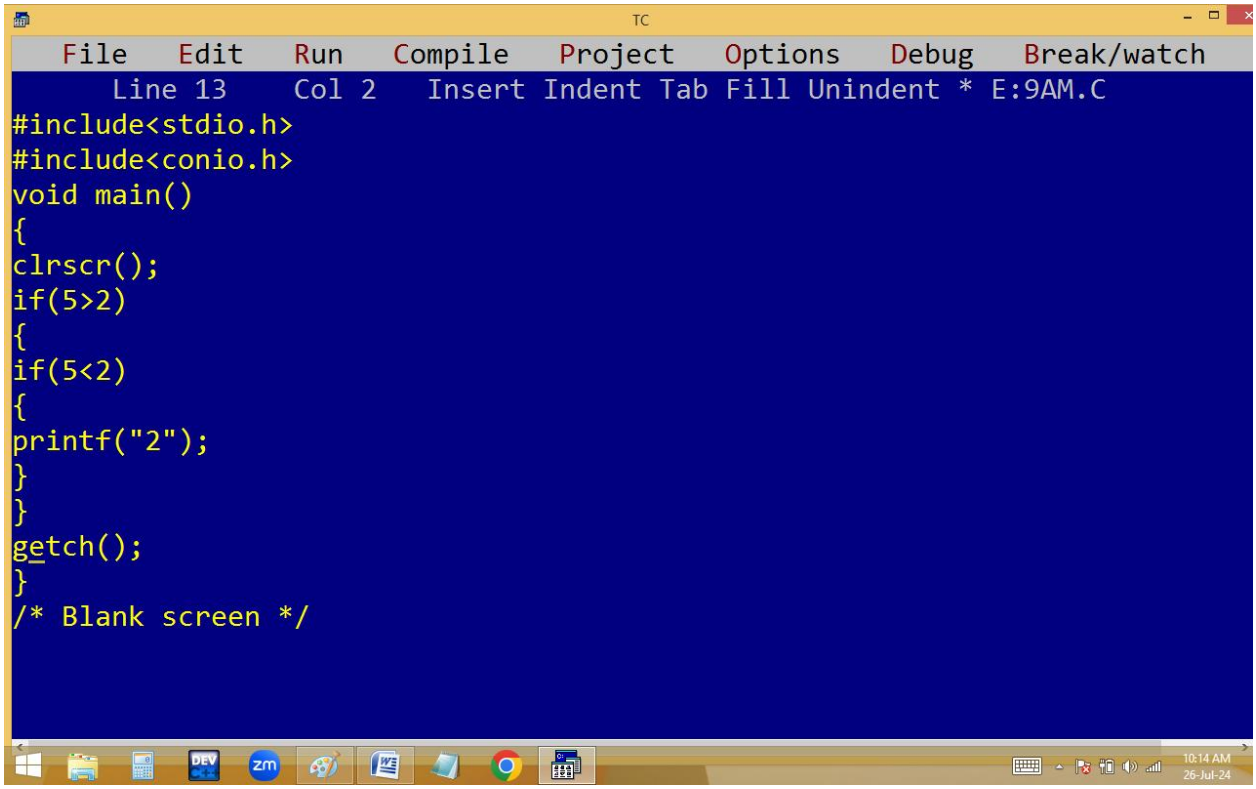
```
if( 0 )
{
if( 1 ) p( 2 );
else  p( 3 );
}
else  p( 4 )
```


TC
File Edit Run Compile Project Options Debug Break/watch
Error: Misplaced else in function main
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
if(5>2)
printf("1");
else printf("2");
else printf("3");
else printf("4");
getch();
}
/* Error */


TC
File Edit Run Compile Project Options Debug Break/watch
Error: Statement missing ; in function main
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
if(5>2)
printf("1");
else (5<2) printf("2");
getch();
}
/* Error */


TC
File Edit Run Compile Project Options Debug Break/watch
Error: Misplaced else in function main
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
if(5>2) printf("1");
else if(5<2)
{
else printf("2");
}
getch();
}
/* Error */


TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 16 Insert Indent Tab Fill Unindent * E:9AM.C
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
if(5>2)
if(5<2)
printf("2");
getch();
}
/* Blank screen_*/



SWITCH

It is a selection statement.

It is used to execute one case of statements from no of cases according to the switch expression value matched with case expression value. In switch the program is jumped to matching case like the go to label.

It is similar to ladder if in working style.

Switch performance is high when compared with ladder if because of it jumps to matching case.

Syntax:

switch(**condition / expression**)

{

case **constexp1:**

statements;

break;

case **constexp2:**

statements;

break;

case **constexpN**:

statements;

break;

[default: statements; **]**

}

Here switch, case, break, default are the keywords.

In between case and case expression / value at least one space should be provided. Otherwise it will become a label.

case expression/value should be a constant integer/char value. i.e. float / string not allowed.

One case contains one expression only.
case expression doesn't contain any separators like , . etc.

case expression should be end with : (colon)

Each case should be separated with break keyword. Otherwise remaining cases also executed.

Duplicate cases not allowed.

default is similar to the else and all cases are failed then default statements are executed. Default is

optional and we can declare it anywhere in our switch.

Outside case expressions not considered in switch.