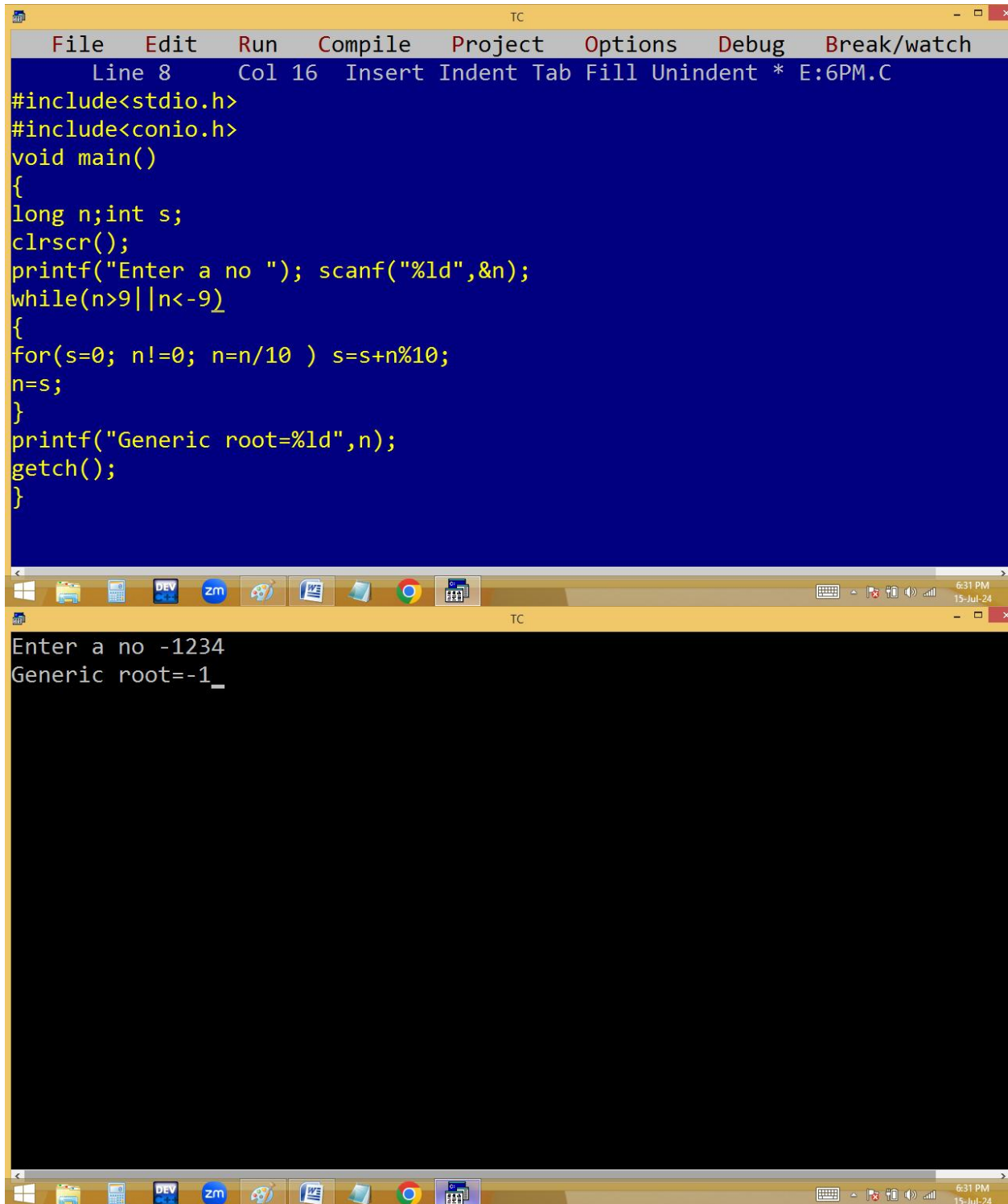


Finding generic root of given no.

$$5677 = 5+6+7+7 = 25 \rightarrow 2+5 = 7$$



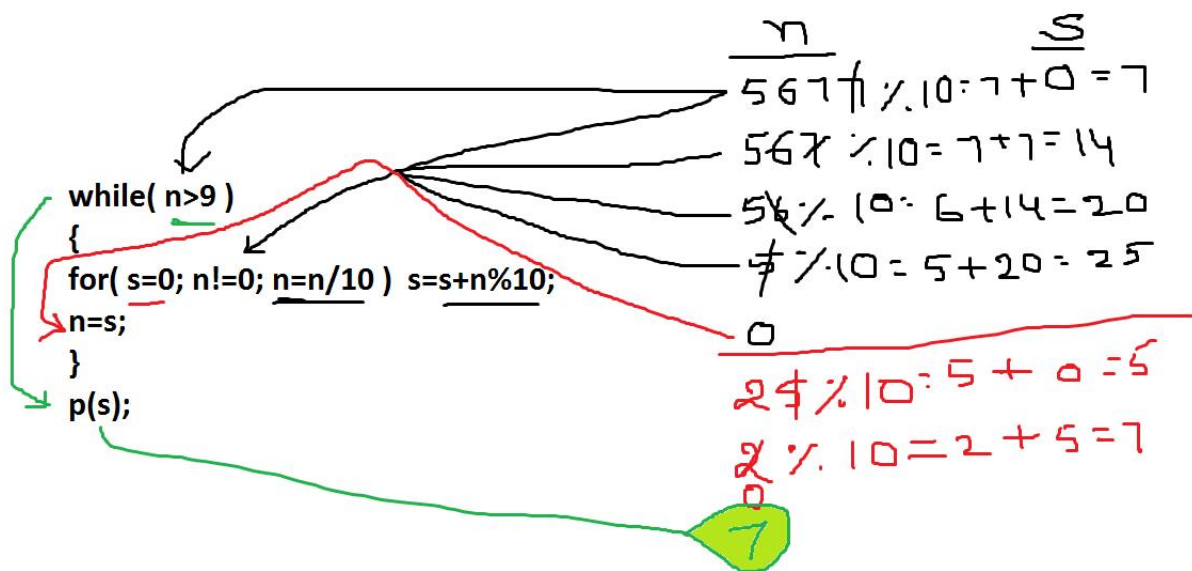
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 16 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
long n;int s;
clrscr();
printf("Enter a no "); scanf("%ld",&n);
while(n>9||n<-9)
{
for(s=0; n!=0; n=n/10 ) s=s+n%10;
n=s;
}
printf("Generic root=%ld",n);
getch();
}
```

Enter a no -1234
Generic root=-1_

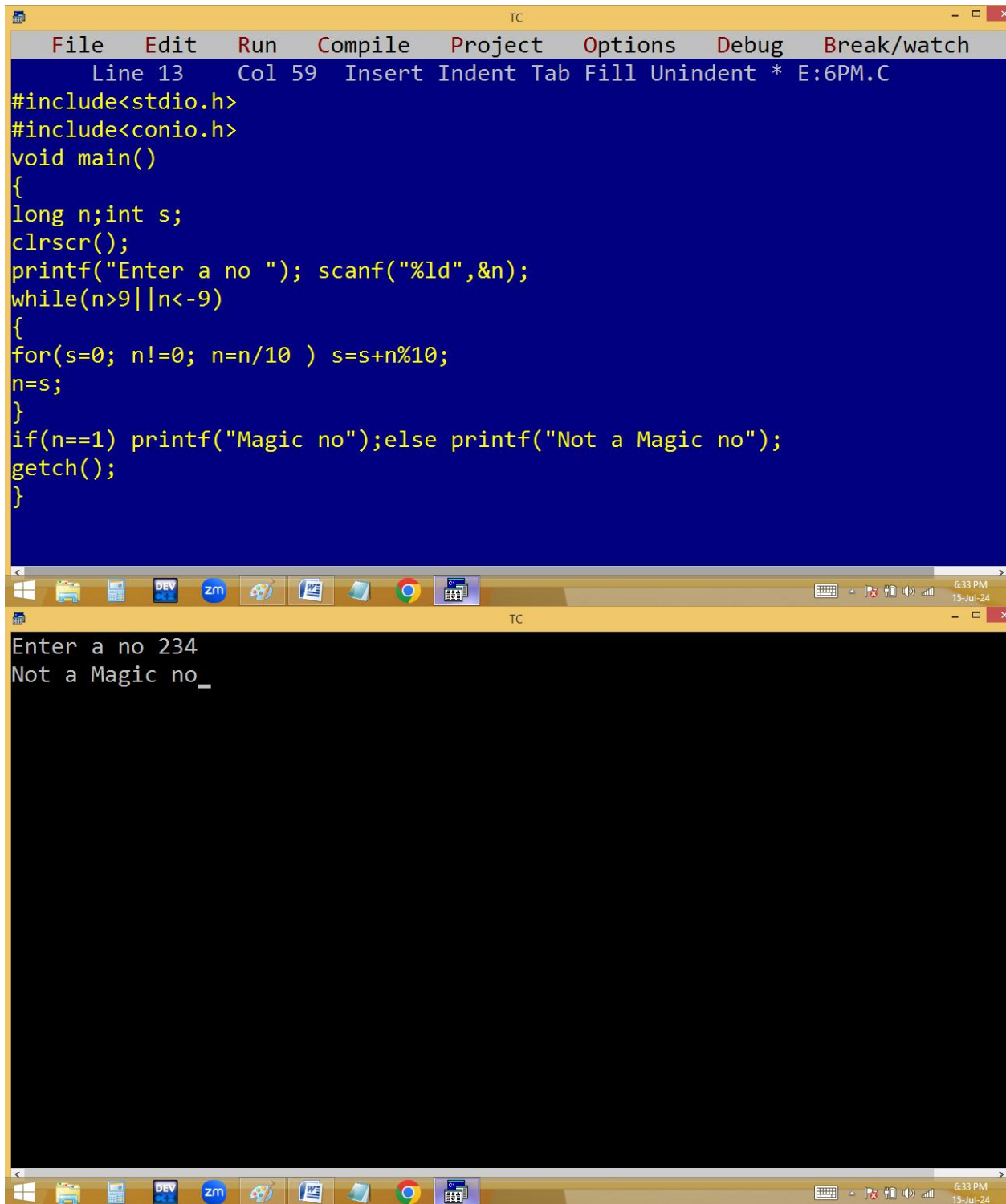
```
TC
Enter a no 1234
Generic root=1
```

```
TC
Enter a no 0
Generic root=0
```

```
Enter a no 567
Generic root=9_
```



1234 → 1+2+3+4 = 10 → 1+0 =



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program for checking magic numbers. The code is as follows:

```
Line 13 Col 59 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
long n;int s;
clrscr();
printf("Enter a no "); scanf("%ld",&n);
while(n>9||n<-9)
{
for(s=0; n!=0; n=n/10 ) s=s+n%10;
n=s;
}
if(n==1) printf("Magic no");else printf("Not a Magic no");
getch();
}
```

The bottom window shows the program's execution. It prompts the user to "Enter a no" and the input "234" is provided. The output is "Not a Magic no_".

Finding strong no or not?

Sum of factorial of each digit is equal to given no.

1 factorial 1

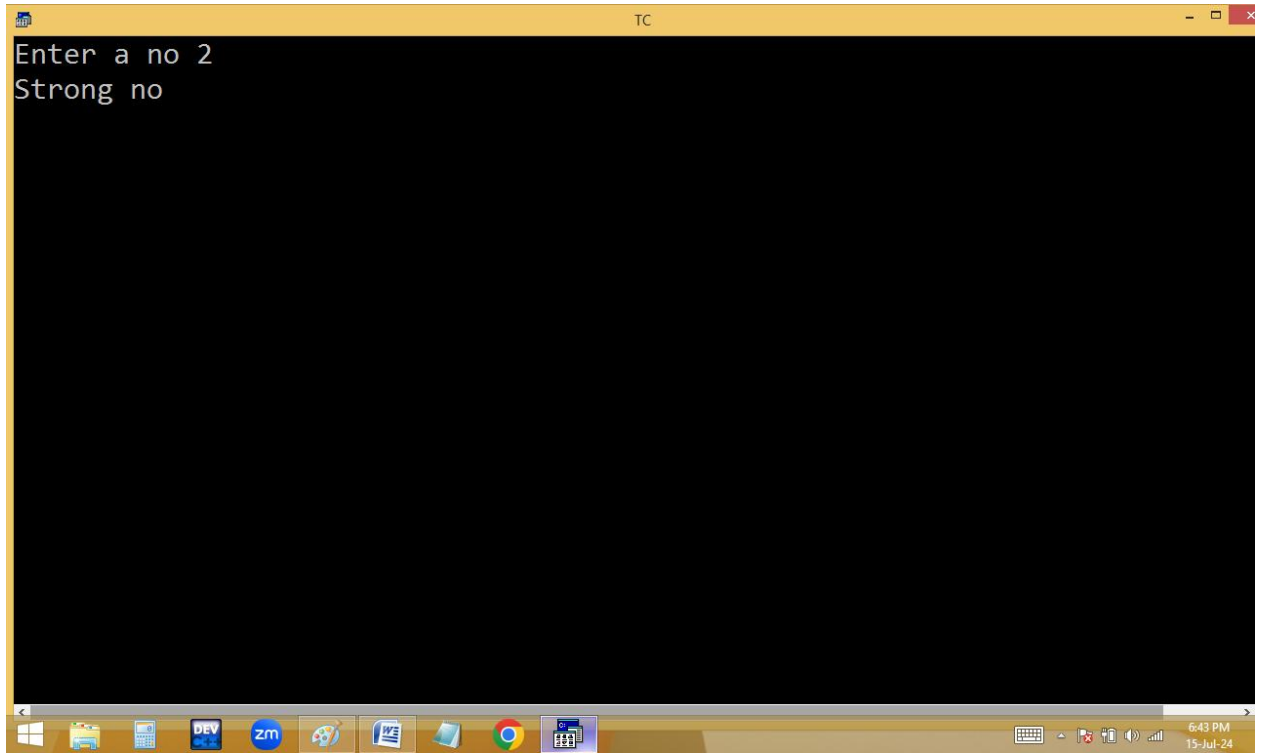
2 factorial 2

3 factorial 6 ← not a strong no

145 → $1! + 4! + 5! = 1 + 24 + 120 = 145$

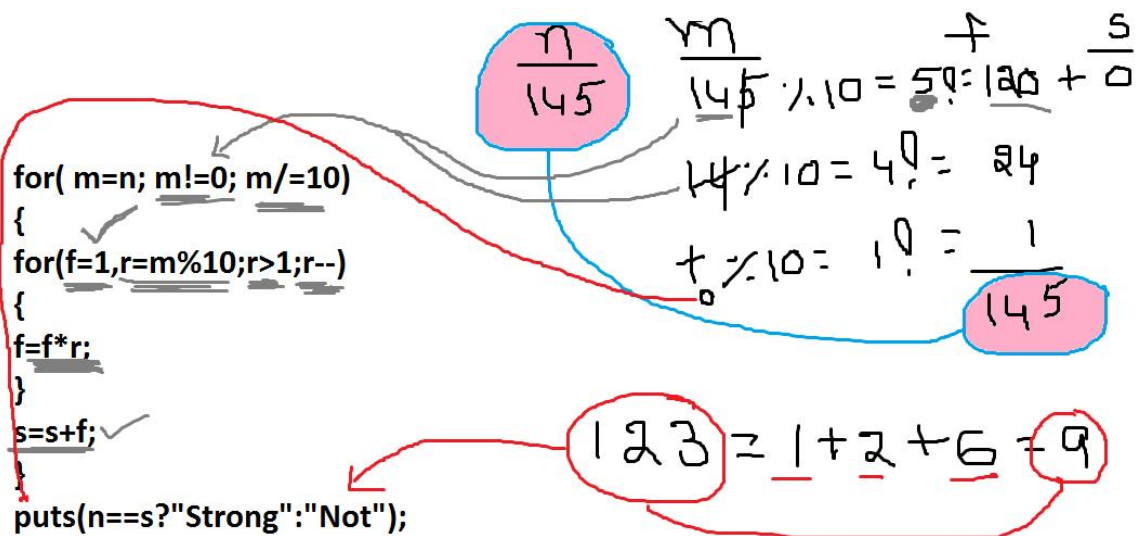
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 55 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,m,f,r,s=0;
clrscr();
printf("Enter a no "); scanf("%d",&n);
for(m=n; m!=0; m=m/10 )
{
for(f=1, r=m%10; r>1 ; r-- ) f=f*r;
s=s+f;
}
if(n==s) printf("Strong no");else printf("Not a Strong_no");
getch();
}
```

```
TC
Enter a no 1
Strong no_
```



```
TC
Enter a no 3
Not a Strong no
```

```
TC
Enter a no 145
Strong no_
```

Printing 1..n palindrome no's and count.

Eg: 1 to 50 palindromes and count

1 to 9 11 22 33 44 – 13 palindromes

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 11 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,a,b,c=0,r,rev;
clrscr();
printf("Enter a no "); scanf("%d",&n);
for(a=1; a<=n; a++ )
{
for(rev=0, b=a; b!=0 ; b=b/10) {r=b%10; rev=rev*10+r;}
if(a==rev) printf("%4d",a,c++);
}
printf("\n%d palindromes",c);
getch();
}
```

```
TC
Enter a no 50
1 2 3 4 5 6 7 8 9 11 22 33 44
13 palindromes_
```

```

Enter a no 500
1 2 3 4 5 6 7 8 9 11 22 33 44 55 66 77 88 99 10
121 131 141 151 161 171 181 191 202 212 222 232 242 252 262 272 282 292 30
323 333 343 353 363 373 383 393 404 414 424 434 444 454 464 474 484 494
58 palindromes_

```

```

for( a=1; a<=n; a++ )
{
  for( b=a, rev=0; b!=0; b=b/10 )
  {
    r=b%10; rev=rev*10+r;
  }
  if(a==rev)p(a, c++);
}
p(c);

```

$\frac{n}{50}$	$\frac{a \rightarrow b}{1}$	$\frac{r}{0 \div 10 = 1}$	$\frac{rev}{0 \times 10 + 1 = 1}$	$\frac{c}{0}$
✓ 1	✓ 2	$2 \div 10 = 2$	$0 \times 10 + 2 = 2$	2
		$2 \div 10 = 0$	$0 \times 10 + 0 = 0$	9
		$1 \div 10 = 1$	$0 \times 10 + 1 = 1$	10
	✓ 11	$11 \div 10 = 1$	$0 \times 10 + 1 = 1$	
		$1 \div 10 = 1$	$1 \times 10 + 1 = 11$	

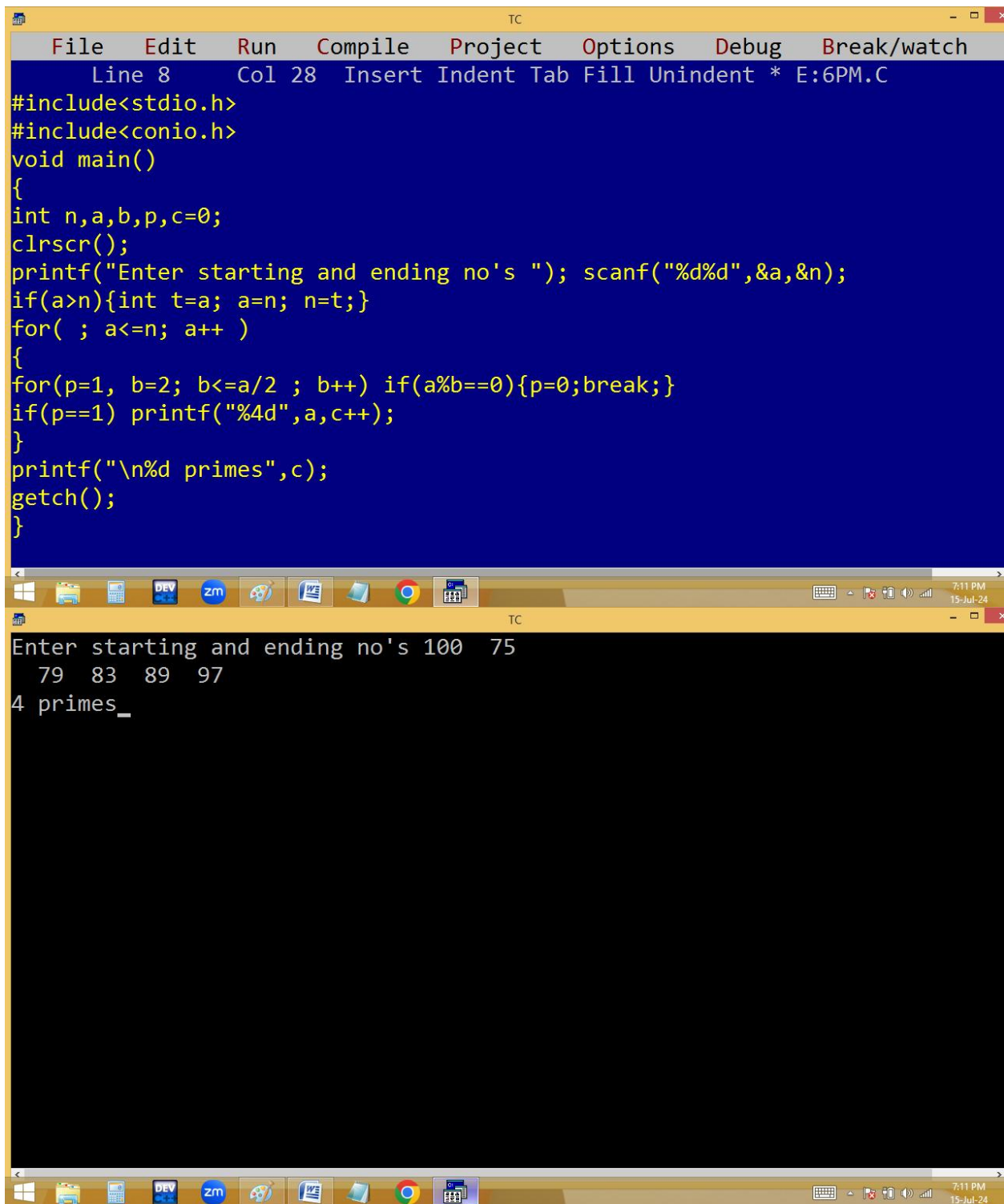
Print 1..n primes and count:

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 1 Col 29 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,a,b,p,c=0;
clrscr();
printf("Enter a no "); scanf("%d",&n);
for(a=2; a<=n; a++ )
{
for(p=1, b=2; b<=a/2 ; b++) if(a%b==0){p=0;break;}
if(p==1) printf("%4d",a,c++);
}
printf("\n%d primes",c);
getch();
}
```

```
TC
Enter a no 10
    2    3    5    7
4 primes
```

```
TC
Enter a no 200
 2  3  5  7 11 13 17 19 23 29 31 37 41 43 47 53 59 61 67
73 79 83 89 97 101 103 107 109 113 127 131 137 139 149 151 157 163 167
179 181 191 193 197 199
46 primes_
```

Print n to n primes and count:



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program designed to find and print prime numbers between two user-input values. The code includes standard headers, declares variables for the range, a counter, and a temporary value. It uses nested loops: an outer loop for the range and an inner loop to check for divisibility. A counter 'c' tracks the number of primes found. The bottom window shows the program's execution, where the user has entered 100 and 75. The output lists the primes 79, 83, 89, and 97, and states there are 4 primes in total.

```
File Edit Run Compile Project Options Debug Break/watch
Line 8 Col 28 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int n,a,b,p,c=0;
clrscr();
printf("Enter starting and ending no's "); scanf("%d%d",&a,&n);
if(a>n){int t=a; a=n; n=t;}
for( ; a<=n; a++ )
{
for(p=1, b=2; b<=a/2 ; b++) if(a%b==0){p=0;break;}
if(p==1) printf("%4d",a,c++);
}
printf("\n%d primes",c);
getch();
}
```

Enter starting and ending no's 100 75
79 83 89 97
4 primes_

```
TC
Enter starting and ending no's 200
500
211 223 227 229 233 239 241 251 257 263 269 271 277 281 283 293 307 311 31
331 337 347 349 353 359 367 373 379 383 389 397 401 409 419 421 431 433 43
449 457 461 463 467 479 487 491 499
49 primes_
```

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 1 Col 19 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h> #include<conio.h>
void main()
{
int n,a,b,p,c=0;
clrscr();
printf("Enter starting and ending no's "); scanf("%d%d",&a,&n);
if(a>n){int t=a; a=n; n=t;}
if(a==1) a=2;
while(a<=n)
{
p=1; b=2; while(b<=a/2){ if(a%b==0){p=0;break;} b++; }
if(p==1) printf("%4d",a,c++);
a++;
}
printf("\n%d primes",c);
getch();
}

Enter starting and ending no's 50 75
53 59 61 67 71 73
6 primes_

Page: 15 of 16 Words: 116 120% 7:15 PM 15-Jul-24
```


break: It is a keyword used within switch / loop.

When break is occurred program execution jumped to the first statement after the switch / loop.

TC

File Edit Run Compile Project Options Debug Break/watch

Error: Misplaced break in function main

```
#include<stdio.h>
#include<conio.h>
void main()
{
clrscr();
puts("Good evening");
break;
puts("Good night");
getch();
}
```

TC

File Edit Run Compile Project Options Debug Break/watch

Line 11 Col 21 Insert Indent Tab Fill Unindent * E:6PM.C

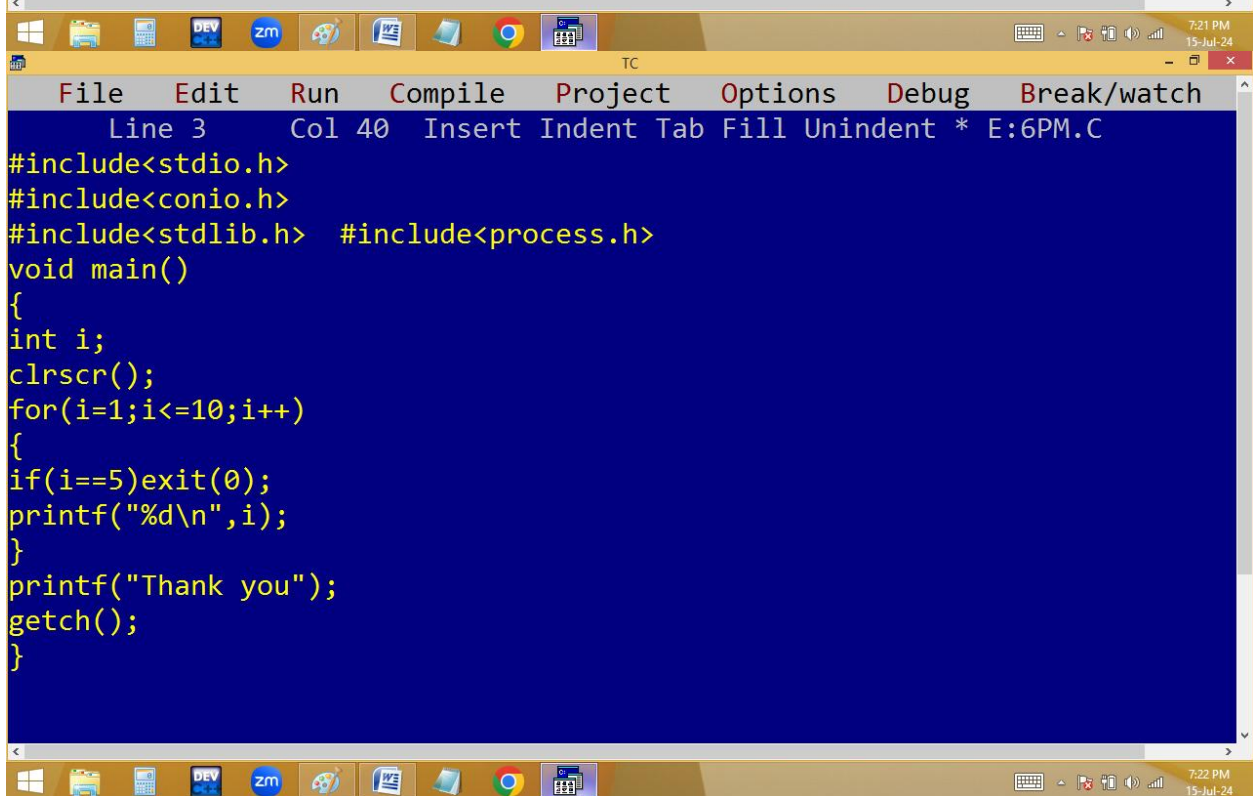
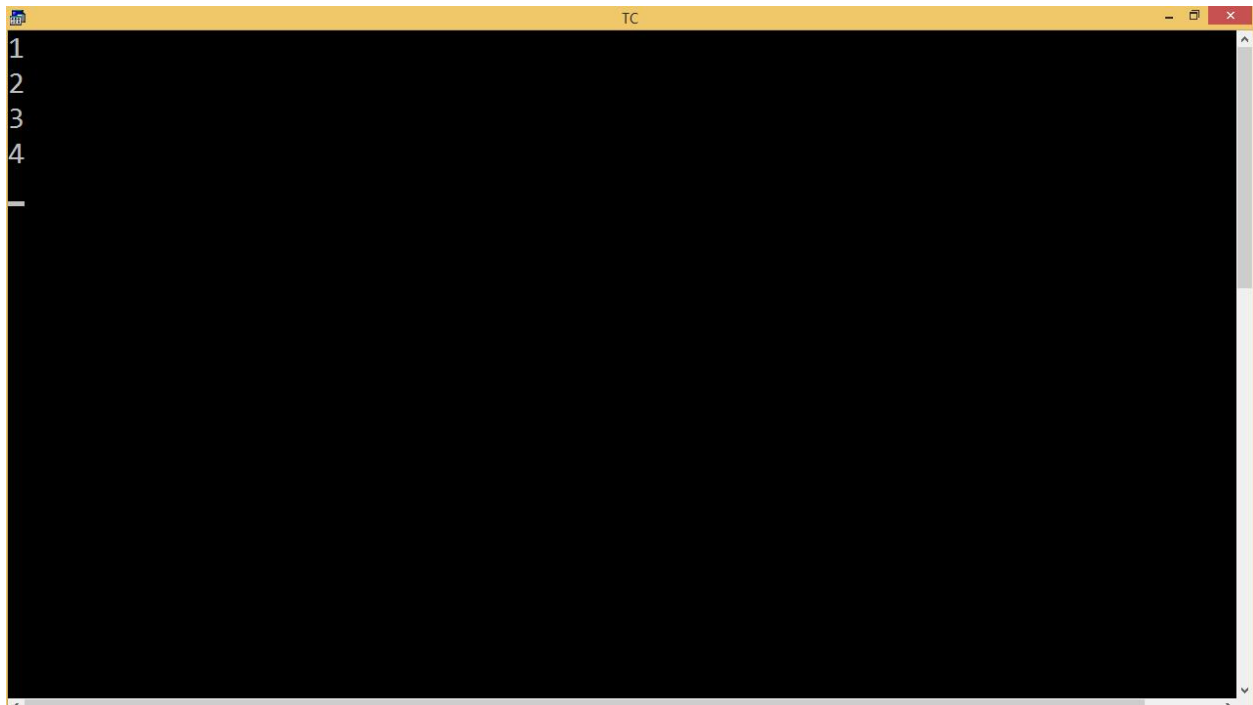
```
#include<stdio.h>
#include<conio.h>
void main()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
printf("%d\n",i);
}
printf("Thank you");_
getch();
}
```

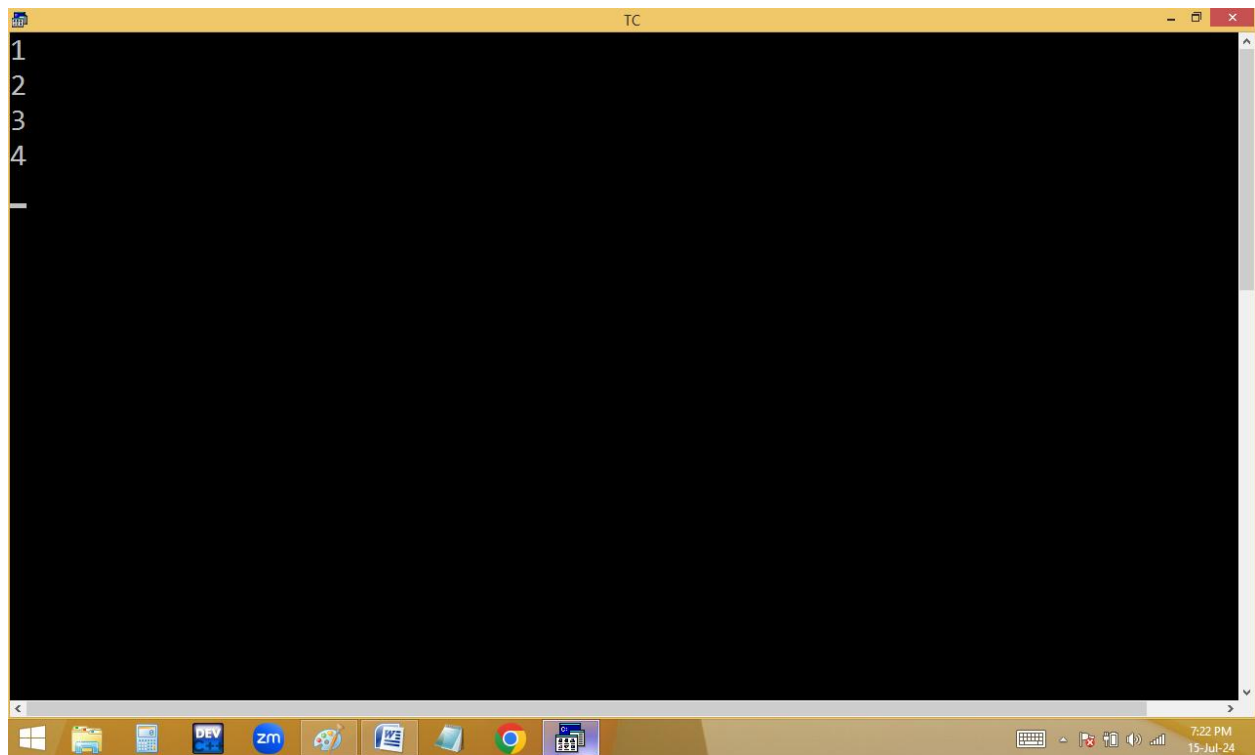
```
1
2
3
4
5
6
7
8
9
10
Thank you
```

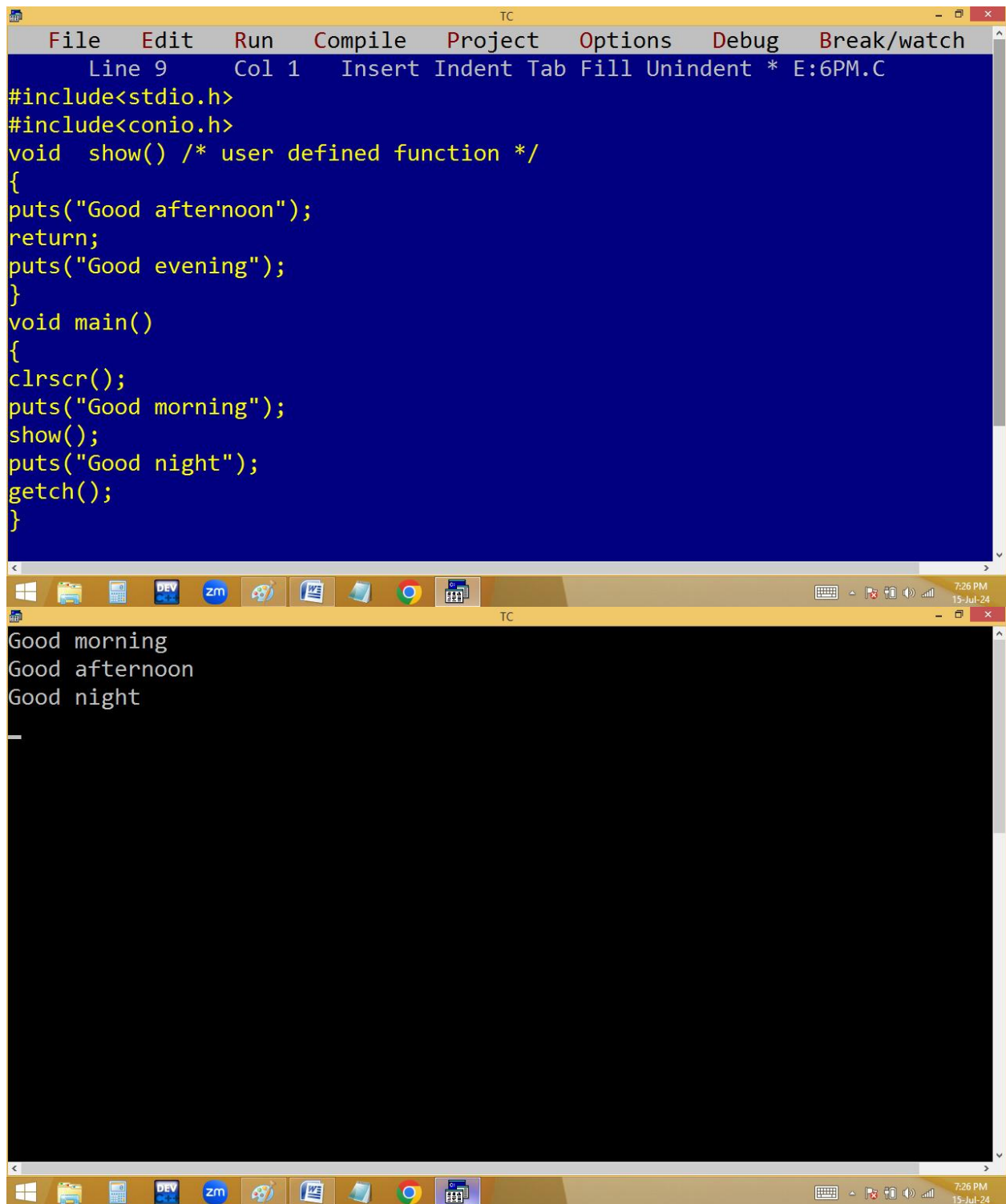
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 15 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
if(i==5)break;
printf("%d\n",i);
}
printf("Thank you");
getch();
}
```

```
TC
1
2
3
4
Thank you
```

```
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()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
if(i==5)return;
printf("%d\n",i);
}
printf("Thank you");
getch();
}
```







The image shows a screenshot of a Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 1 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
void show() /* user defined function */
{
puts("Good afternoon");
return;
puts("Good evening");
}
void main()
{
clrscr();
puts("Good morning");
show();
puts("Good night");
getch();
}
```

The bottom window shows the output of the program:

```
Good morning
Good afternoon
Good night
```

The Windows taskbar at the bottom includes icons for the Start menu, File Explorer, Task View, DEV, ZOOM, and other applications. The system clock in the bottom right corner indicates 7:26 PM on 15-Jul-24.

The screenshot shows the Turbo C++ (TC) IDE. The top window displays the source code for a C program named E:6PM.C. The code includes headers for stdio.h, conio.h, and stdlib.h. It defines a function 'show()' that prints 'Good afternoon' and 'Good evening', and a 'main()' function that prints 'Good morning', calls 'show()', prints 'Good night', and waits for a key press using 'getch()'. The bottom window shows the program's output, which is 'Good morning' followed by 'Good afternoon' on the next line. The Windows taskbar at the bottom shows the time as 7:27 PM on 15-Jul-24.

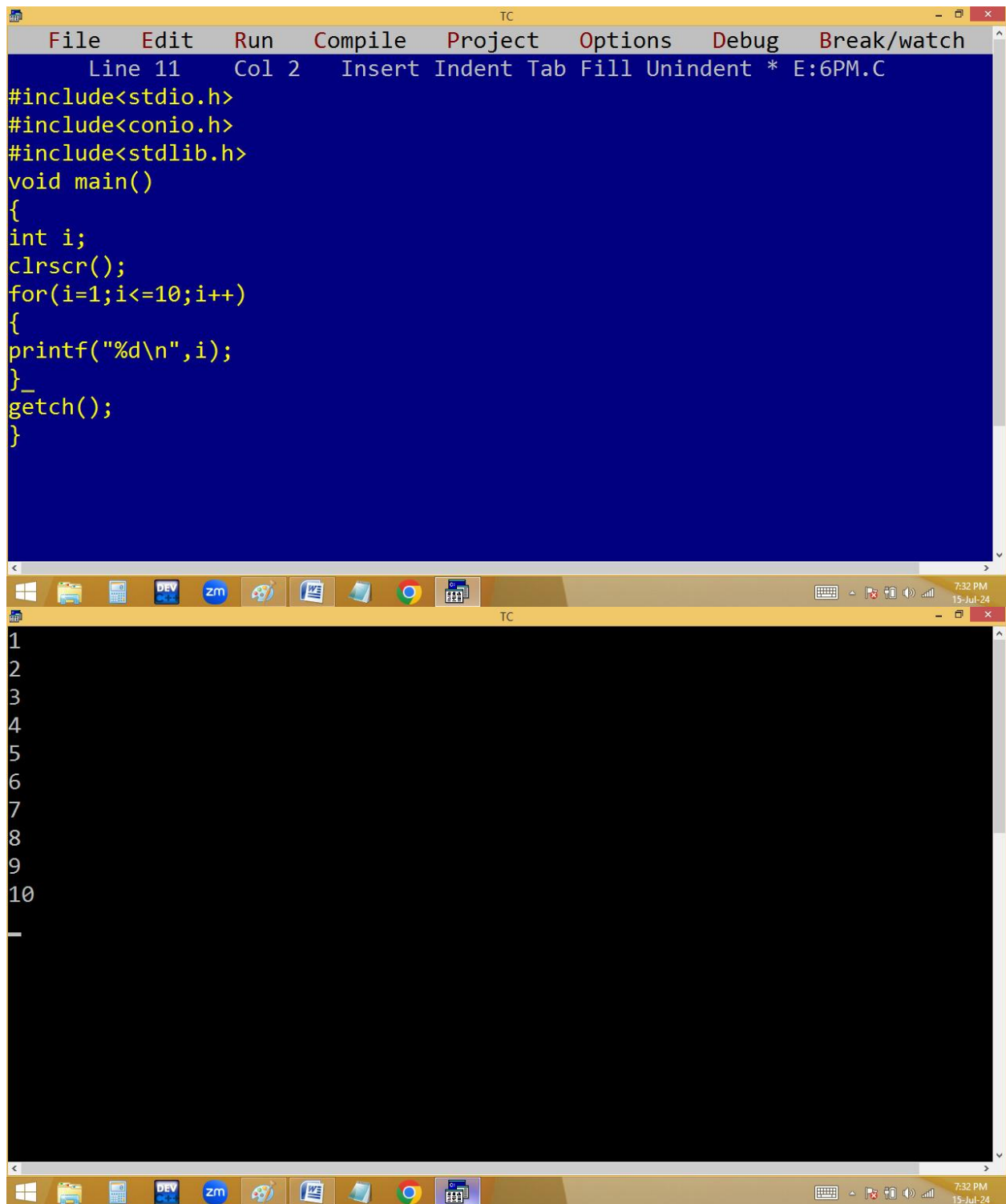
```
File Edit Run Compile Project Options Debug Break/watch
Line 3 Col 19 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void show() /* user defined function */
{
puts("Good afternoon");
exit(0);
puts("Good evening");
}
void main()
{
clrscr();
puts("Good morning");
show();
puts("Good night");
getch();
}
```

Good morning
Good afternoon

break	exit()	Return
Keyword	Function	Keyword
Header file not required	stdlib.h / process.h required	Header file not required

Terminates switch / loop	Terminates total program	Closes that function only
Should be used within switch/loop	Used any where	Used any where

Continue: when continue word is occurred, the program execution jumped to end of the loop and later beginning of the loop. Due to this the remaining statements in that loop are not working. When we want to skip the statements in a loop then only use the continue keyword.



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

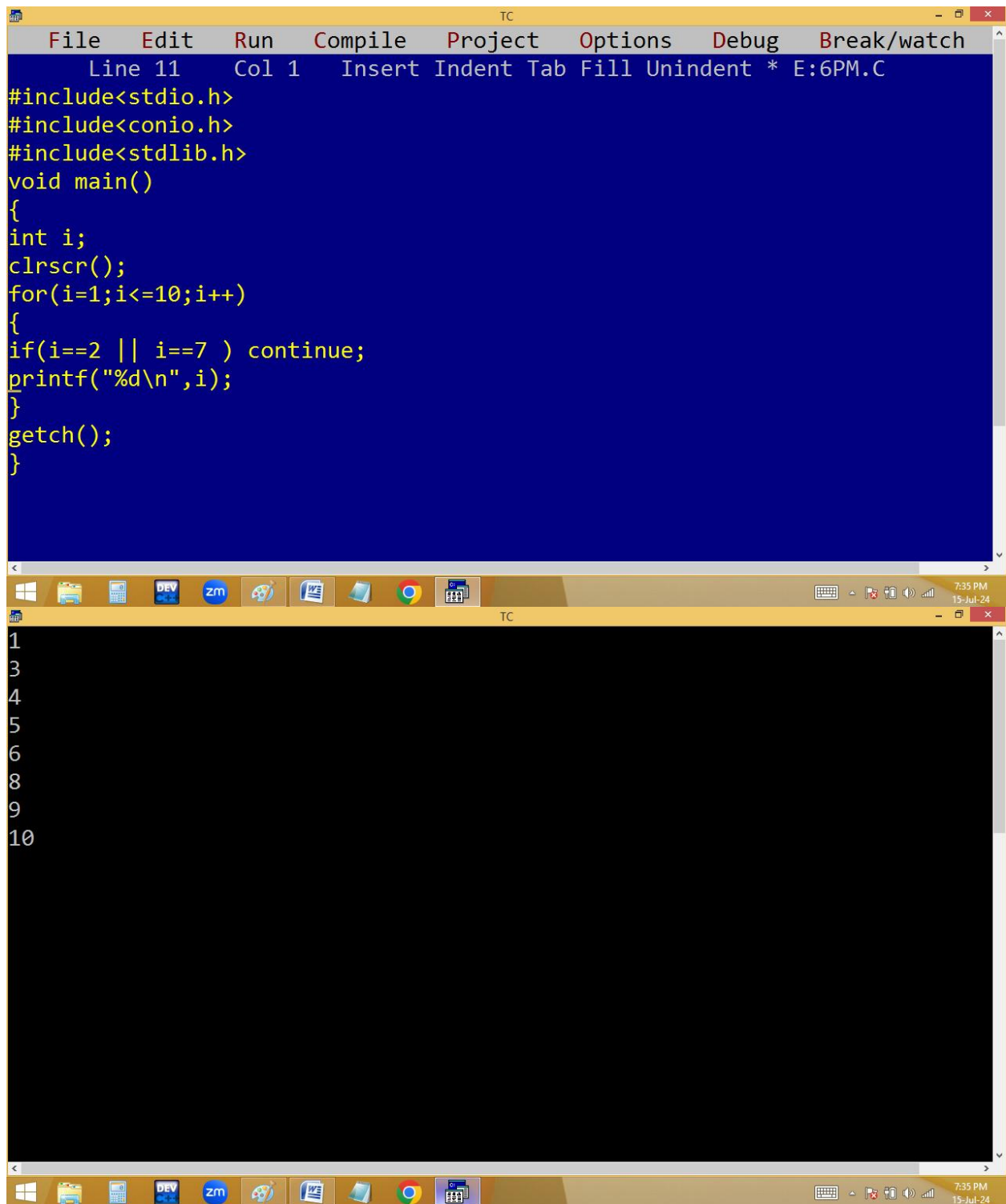
```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 2 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
printf("%d\n",i);
}_
getch();
}
```

The bottom window shows the output of the program, which is a list of numbers from 1 to 10, each on a new line:

```
1
2
3
4
5
6
7
8
9
10
_
```

The Windows taskbar at the bottom shows the time as 7:32 PM on 15-Jul-24.

Printing 1 to 10 except 2 and 10.



The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program that prints odd numbers from 1 to 10 using the `continue` statement. The code is as follows:

```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 1 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
    int i;
    clrscr();
    for(i=1;i<=10;i++)
    {
        if(i==2 || i==7 ) continue;
        printf("%d\n",i);
    }
    getch();
}
```

The bottom window shows the output of the program, which is the sequence of odd numbers from 1 to 10, each on a new line:

```
1
3
4
5
6
8
9
10
```

The taskbar at the bottom of the screen shows various application icons, including Windows Explorer, DEV C++, Zoom, and Google Chrome. The system clock in the bottom right corner indicates the time is 7:35 PM on 15-Jul-24.

Printing 1..10 odd numbers using continue:

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 10 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
if(i%2==0_) continue;
printf("%d\n",i);
}
getch();
}
```

1
3
5
7
9

Even numbers:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 8 Insert Indent Tab Fill Unindent * E:6PM.C
#include<stdio.h>
#include<conio.h>
#include<stdlib.h>
void main()
{
int i;
clrscr();
for(i=1;i<=10;i++)
{
if(i%2!=0 ) continue;
printf("%d\n",i);
}
getch();
}
```

The bottom window shows the output of the program, which is a list of even numbers from 2 to 10, each on a new line:

```
2
4
6
8
10
_
```

The Windows taskbar at the bottom shows the time as 7:36 PM and 7:37 PM on 15-Jul-24.

