

Arranging array elements in reverse order:

The image shows two windows of the Turbo C++ (TC) IDE. The top window displays the source code for a C program that reverses an array. The bottom window shows the program's execution output.

Top Window (Source Code):

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 13 Col 65 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<n/2;i++)
{
t=a[i]; a[i]=a[n-i-1]; a[n-i-1]=t;
}
printf("Reversed elements"); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}
```

Activate Windows
Go to PC settings to activate Windows.

Windows taskbar: 2:30 PM, 26-Sep-23

Bottom Window (Execution Output):

```
TC
Enter array size 1-100 4
Enter 4 elements 9 3 5 7
Reversed elements    7    5    3    9
```

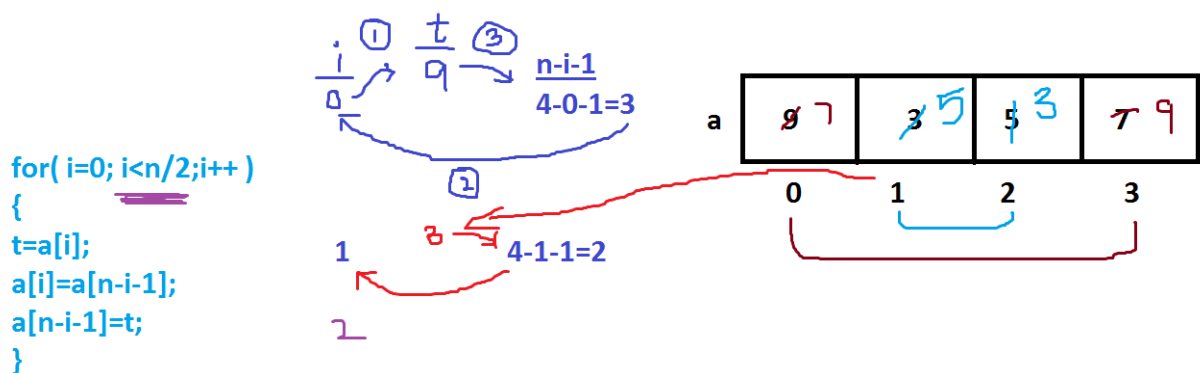
Activate Windows
Go to PC settings to activate Windows.

Windows taskbar: 2:30 PM, 26-Sep-23

```
TC
Enter array size 1-100 5
Enter 5 elements 1 2 3 4 5
Reversed elements 5 4 3 2 1_

Activate Windows
Go to PC settings to activate Windows.

2:31 PM
26-Sep-23
```



Without using 3rd variable:

The image shows two screenshots of the Turbo C++ (TC) IDE. The top screenshot displays the source code for a C program that reverses an array. The code includes headers for `stdio.h` and `conio.h`, and defines a `main` function. Inside `main`, it declares an array `a` of size 100, prompts the user for the array size `n` (with a range of 1-100), and then for `n` elements. It uses a loop to read the elements into the array. Another loop then reverses the array by swapping elements at indices `i` and `n-i-1`. Finally, it prints the reversed elements. The bottom screenshot shows the program's execution. The user enters 5 for the array size and 2 0 1 7 3 for the elements. The output shows the reversed elements as 3 7 1 0 2. Both screenshots include a Windows taskbar at the bottom with various application icons and a system clock showing 2:32 PM and 2:33 PM on 26-Sep-23.

```
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 63 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<n/2;i++)
{
a[i]=a[i]+a[n-i-1]; a[n-i-1]=a[i]-a[n-i-1]; a[i]=a[i]-a[n-i-1];
}
printf("Reversed elements"); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}
```

Enter array size 1-100 5
Enter 5 elements 2 0 1 7 3
Reversed elements 3 7 1 0 2

```

for(i=0;i<n/2;i++)
{
a[i]=a[i]+a[n-i-1];
a[n-i-1]=a[i]-a[n-i-1];
a[i]=a[i]-a[n-i-1];
}

```

i
0

	$16-9=7$	$8-3=5$		
	$9+7=16$	$3+5=8$	$8-5=3$	$16-7=9$
a	9	3	5	7
	0	1	2	3

Linear search:

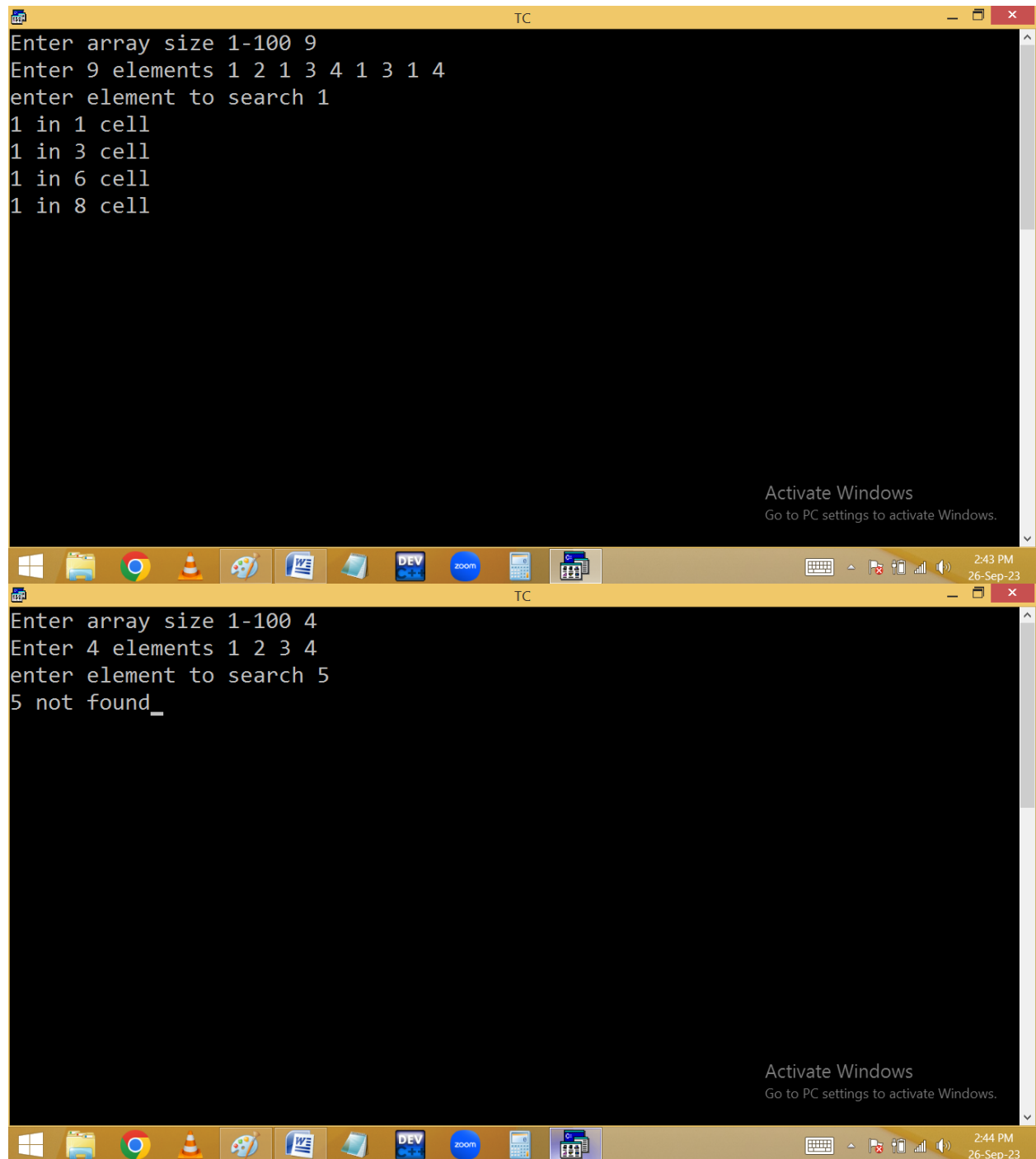
The image shows two windows of the Turbo C++ (TC) IDE. The top window displays the source code for a C program named E:2PM.C. The code is as follows:

```
File Edit Run Compile Project Options Debug Break/watch
Line 15 Col 1 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,ele, f=0; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
printf("enter element to search "); scanf("%d",&ele);
for(i=0;i<n;i++)
{
if(a[i]==ele)printf("%d in %d cell\n",ele,i+1,f=1);
}
if(f==0)printf("%d not found",ele);
getch();
}
```

The bottom window shows the execution output of the program:

```
Enter array size 1-100 5
Enter 5 elements 1 9 2 5 3
enter element to search 2
2 in 3 cell
```

Both windows have a taskbar at the bottom with various application icons and a system tray showing the time as 2:43 PM on 26-Sep-23. An "Activate Windows" watermark is visible in the bottom right of each window.



```

for( i=0; i<4;i++ )
{
if( a[i]==ele )
{
p("%d in %d cell\n", ele, i+1, f=1);
} 5 in 3 cell
}
}

if(f==0) p("element not found");

```

a

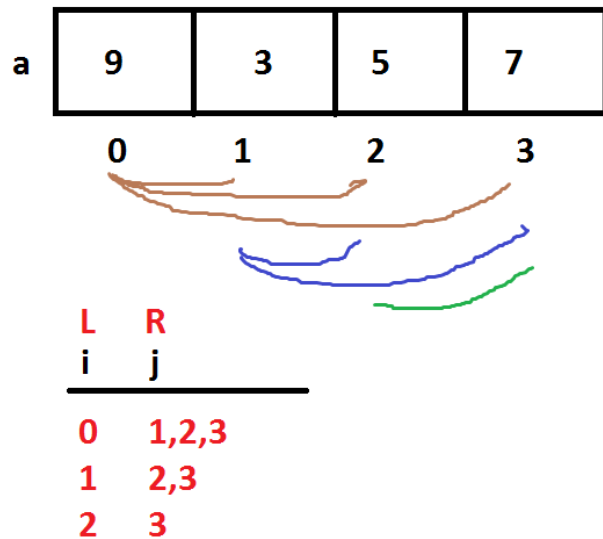
9	3	5	7
0	1	2	3

i	ele	f
0	5	0
1		1
2+1		

Selection sort:

Sorting means arranging data in ascending or descending order.

9	3	5	7
3	9	5	7
3	5	9	7
3	5	7	9



Ascending order:

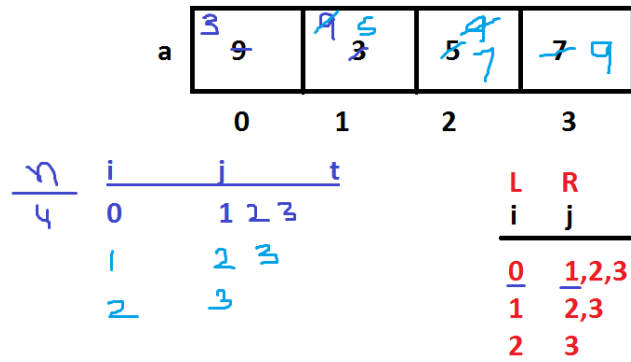

```
TC
Line 18 Col 2 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=i+1;j<=n-1;j++)
{
if(a[i]>a[j]){t=a[i];a[i]=a[j];a[j]=t; }
}
}
printf("Sorted elements "); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}_
```

Enter array size 1-100 9
Enter 9 elements 5 8 2 -6 1 0 5 -4 3
Sorted elements -6 -4 0 1 2 3 5 5 8

```

for( i=0; i<=n-2;i++)
{
for( j=i+1; j<=n-1; j++)
{   L   R
if(a[i]>a[ j] )
{
t=a[i];
a[i]=a[j];
a[j]=t;
}}
}

```



Descending order:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code for a bubble sort program. The code includes headers for `stdio.h` and `conio.h`, and defines a `main` function. Inside `main`, it declares an array `a` of size 100, and variables `n`, `i`, `j`, and `t`. It prompts the user to enter an array size (1-100) and then `n` elements. It then implements the bubble sort algorithm using nested `for` loops and a temporary variable `t` for swapping. Finally, it prints the sorted elements. The bottom window shows the program's execution. It prompts for the array size (7) and the elements (5 0 1 8 -4 8 2). The output shows the sorted elements: 8 8 5 2 1 0 -4. The Windows taskbar at the bottom shows the time as 3:02 PM on 26-Sep-23.

```
Line 18 Col 9 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=i+1;j<=n-1;j++)
{
if(a[i]<a[j]){t=a[i];a[i]=a[j];a[j]=t; }
}
}
printf("Sorted elements "); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}
```

Enter array size 1-100 7
Enter 7 elements 5 0 1 8 -4 8 2
Sorted elements 8 8 5 2 1 0 -4

Bubble sort:

9 3 5 7
 3 9 5 7
 3 5 9 7
 3 5 7 9

a

9	3	5	7
---	---	---	---

0 1 2 3


i	j,j+1
0	0-1, 1-2, 2-3
1	0-1, 1-2
2	0-1

```
TC
Line 18 Col 41 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=0;j<n-i-1;j++)
{
if(a[j]>a[j+1]){t=a[j];a[j]=a[j+1];a[j+1]=t; }
}
}
printf("Sorted elements "); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}
```

Enter array size 1-100 8
Enter 8 elements 4 0 7 -4 8 4 1 9
Sorted elements -4 0 1 4 4 7 8 9

```

for( i=0; i<=n-2;i++)
{
    for( j=0; j<n-i-1; j++ )
    {
        L      R
        if(a[j]>a[j+1])
        {
            t=a[j];a[j]=a[j+1];a[j+1]=t;
        }
    }
}

```

a

3 9	5 9 3	9 1 5	7 9
0	1	2	3

$\frac{i}{0}$ $\frac{j}{0}$
 $\frac{1}{2}$

$n-i-1$
 $4-0-1=3$

i	j,j+1
0	<u>0-1</u> , <u>1-2</u> , <u>2-3</u>
1	<u>0-1</u> , <u>1-2</u>
2	0-1

Descending order:

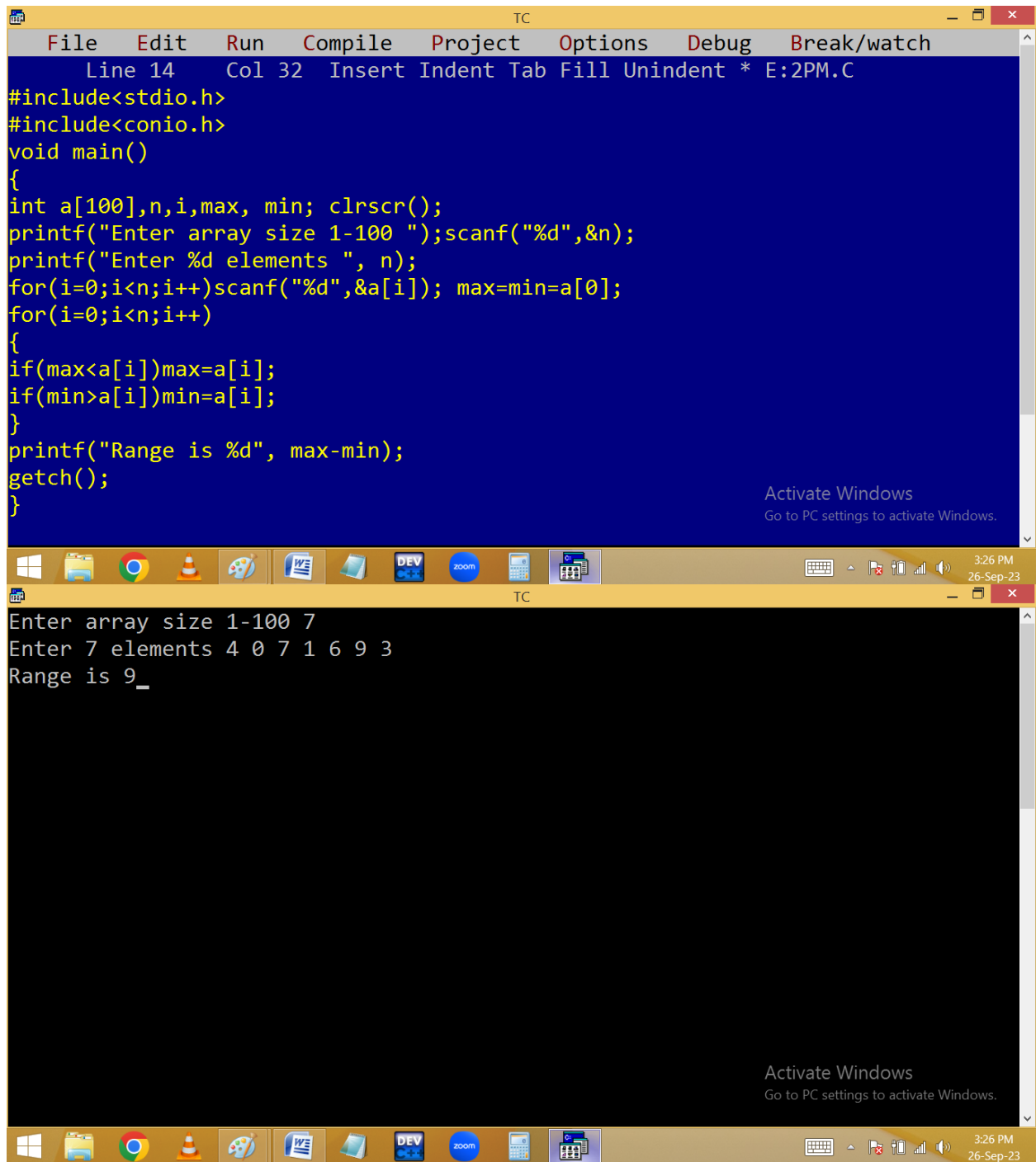
```
TC
Line 18 Col 9 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=0;j<n-i-1;j++)
{
if(a[j]<a[j+1]){t=a[j];a[j]=a[j+1];a[j+1]=t; }
}
}
printf("Sorted elements "); for(i=0;i<n;i++)printf("%4d",a[i]);
getch();
}

TC
Enter array size 1-100 5
Enter 5 elements 2 0 4 1 4
Sorted elements 4 4 2 1 0

Activate Windows
Go to PC settings to activate Windows.
3:23 PM
26-Sep-23
```

Finding the range of unsorted array:

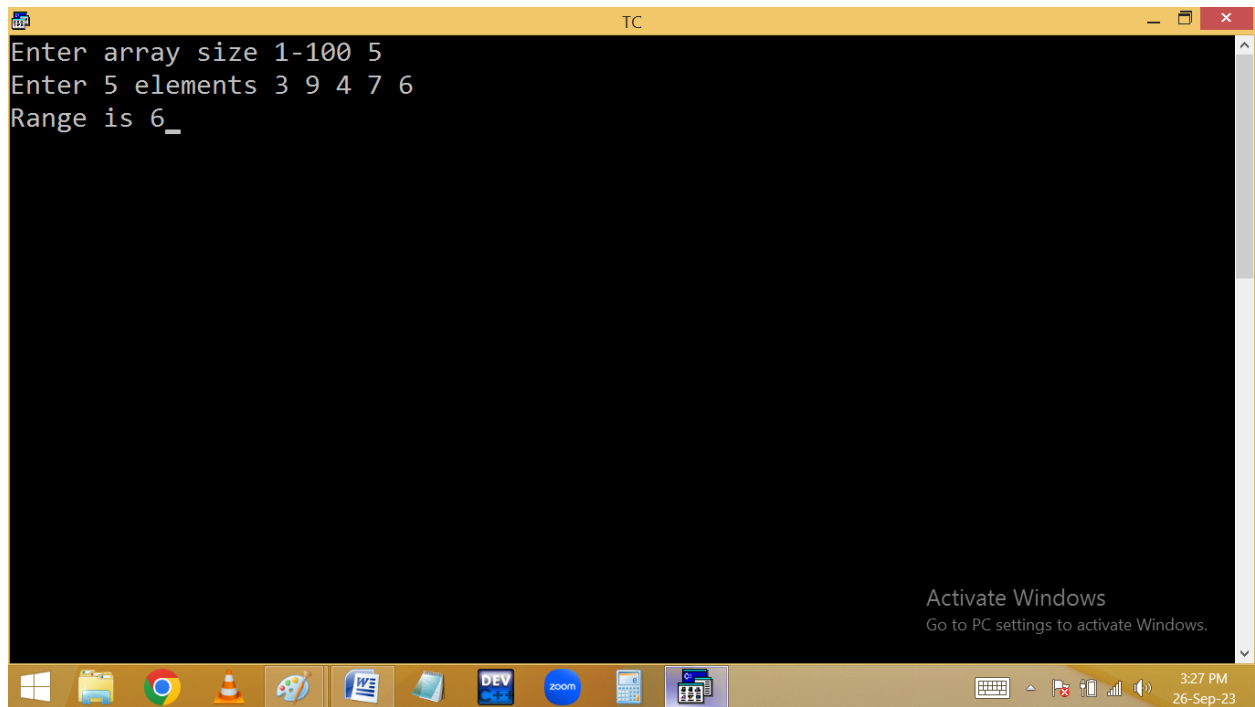
Max-min=range



The image shows two windows of the Turbo C++ (TC) IDE. The top window displays the source code for a C program named E:2PM.C. The code calculates the range of an array by finding the maximum and minimum values. The bottom window shows the program's execution output, where the user has entered an array size of 7 and elements 4, 0, 7, 1, 6, 9, 3, resulting in a range of 9.

```
File Edit Run Compile Project Options Debug Break/watch
Line 14 Col 32 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,max, min; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]); max=min=a[0];
for(i=0;i<n;i++)
{
if(max<a[i])max=a[i];
if(min>a[i])min=a[i];
}
printf("Range is %d", max-min);
getch();
}
```

Enter array size 1-100 7
Enter 7 elements 4 0 7 1 6 9 3
Range is 9_

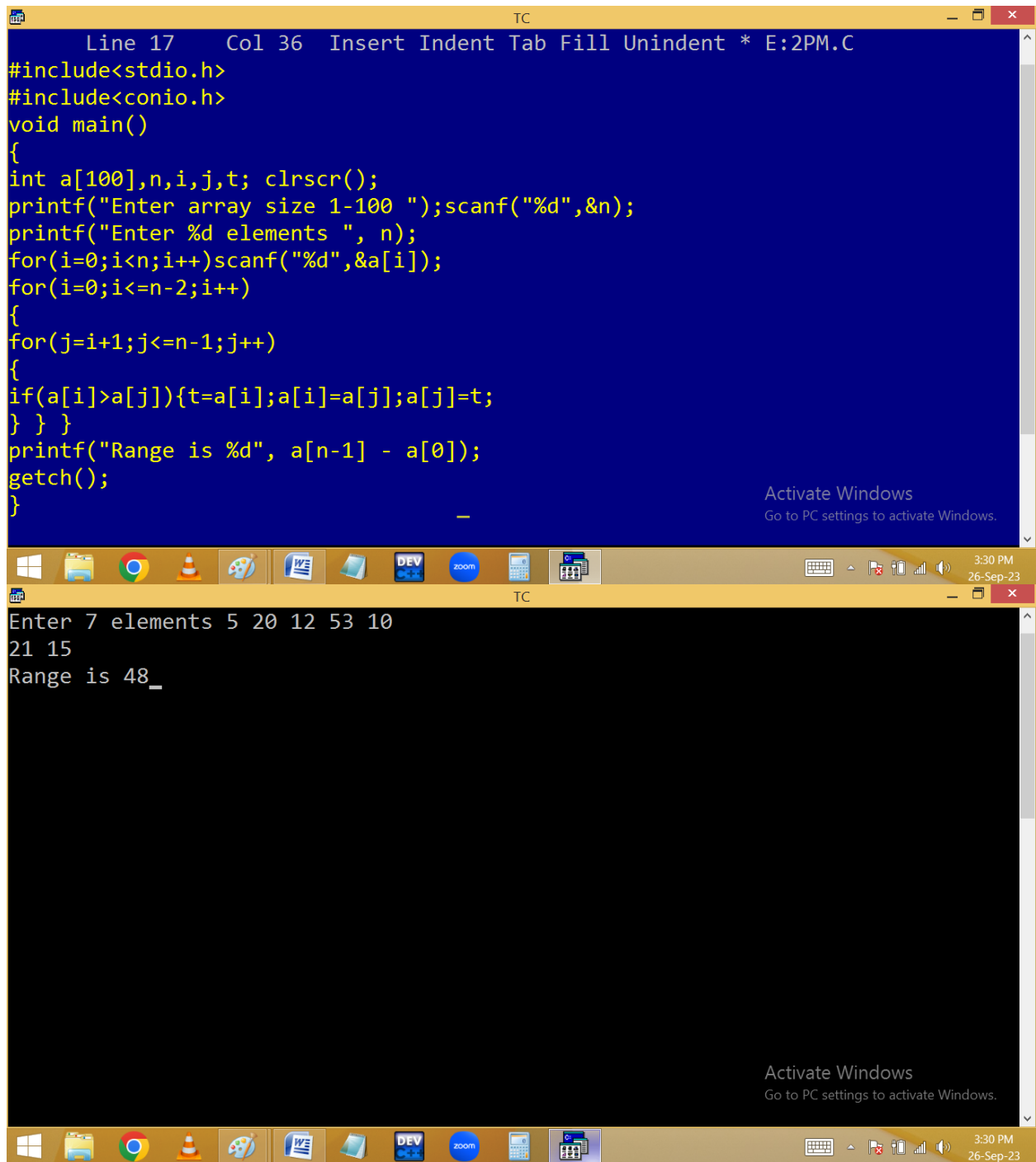


```
TC
Enter array size 1-100 5
Enter 5 elements 3 9 4 7 6
Range is 6_

Activate Windows
Go to PC settings to activate Windows.
```

The screenshot shows a Turbo C++ (TC) window with a black background. The text input shows the user has entered an array size of 5 and five elements: 3, 9, 4, 7, and 6. The program has calculated the range of the sorted array (3, 4, 6, 7, 9) as 6. The Windows taskbar at the bottom shows various icons including File Explorer, Chrome, VLC, Paint, Word, and several development tools. The system clock indicates 3:27 PM on 26-Sep-23.

Finding range of sorted array:



```
Line 17 Col 36 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=i+1;j<=n-1;j++)
{
if(a[i]>a[j]){t=a[i];a[i]=a[j];a[j]=t;
} } }
printf("Range is %d", a[n-1] - a[0]);
getch();
}
```

Enter 7 elements 5 20 12 53 10
21 15
Range is 48_

Find the median of sorted array:

```
TC
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t,m; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=i+1;j<=n-1;j++)
{
if(a[i]>a[j]){t=a[i];a[i]=a[j];a[j]=t;
} } }
m=a[n/2];
printf("Sorted elements ");for(i=0;i<n;i++)printf("%4d",a[i]);
if(n%2==0) m=(m+a[n/2-1])/2;
printf("\nMedian is %d", m);
getch(); }
```

Enter array size 1-100 5
Enter 5 elements 5 -2 9 1 4
Sorted elements -2 1 4 5 9
Median is 4_

```
TC
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; float m; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
for(i=0;i<=n-2;i++)
{
for(j=i+1;j<=n-1;j++)
{
if(a[i]>a[j]){t=a[i];a[i]=a[j];a[j]=t;
} } }
m=a[n/2];
printf("Sorted elements ");for(i=0;i<n;i++)printf("%4d",a[i]);
if(n%2==0) m=(m+a[n/2-1])/2.0;
printf("\nMedian is %f", m);
getch(); }
```

Enter array size 1-100 4
Enter 4 elements 1 2 3 4
Sorted elements 1 2 3 4
Median is 2.500000

Median of unsorted array:

The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays the source code for a C program named E:2PM.C. The code calculates the median of an array of size n. The program prompts the user to enter the array size (1-100) and then the elements. It then calculates the median and displays the result. The bottom window shows the program's execution output.

```
File Edit Run Compile Project Options Debug Break/watch
Line 10 Col 1 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],n,i,j,t; float m; clrscr();
printf("Enter array size 1-100 ");scanf("%d",&n);
printf("Enter %d elements ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
m=a[n/2];
if(n%2==0) m=(m+a[n/2-1])/2.0;
printf("\nMedian is %f", m);
getch(); }
```

Enter array size 1-100 5
Enter 5 elements 1 8 4 2 9

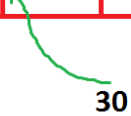
Median is 4.000000

```
TC
Enter array size 1-100 4
Enter 4 elements 1 7 2 9

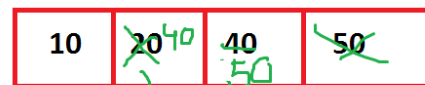
Median is 4.500000
```

Activate Windows
Go to PC settings to activate Windows.

3:44 PM
26-Sep-23



inserting new element in specified pos of array
30 in 3rd cell



deleting particular element
from array

