

ARRAYS

It is collection of homogeneous [same type] variables.

Array is nothing but collection of contiguous memory locations, where we can store and manage more than one value of same type under one name.

It is a derived data type.

It is an implicit / internal pointer.

It is a implicit const pointer

It is one of data structure.

Advantages:

Generally to store several values of same type, we have to declare several variables. Here we have to remember all these variable names also. When the program is too big, it is very difficult to remember all the variable names. In this situation, the only solution is array.

Array reduce program length.

Array minimize the errors.

In functions to carry several values of same type at a time, we are using arrays.

It allows to arrange our data in a order.

Disadvantage:

Array size is Constant Positive Integer value. Due to this we are not able to change the array size at run time. Sometimes it causes memory wastage / shortage.

In C language we are using

1. One dimensional arrays
2. Multi dimensional arrays

One dimensional arrays:

- An array with one row and several columns.
- An array with single subscripting operator **[]** is called one dimensional array.
- It is an implicit single pointer.

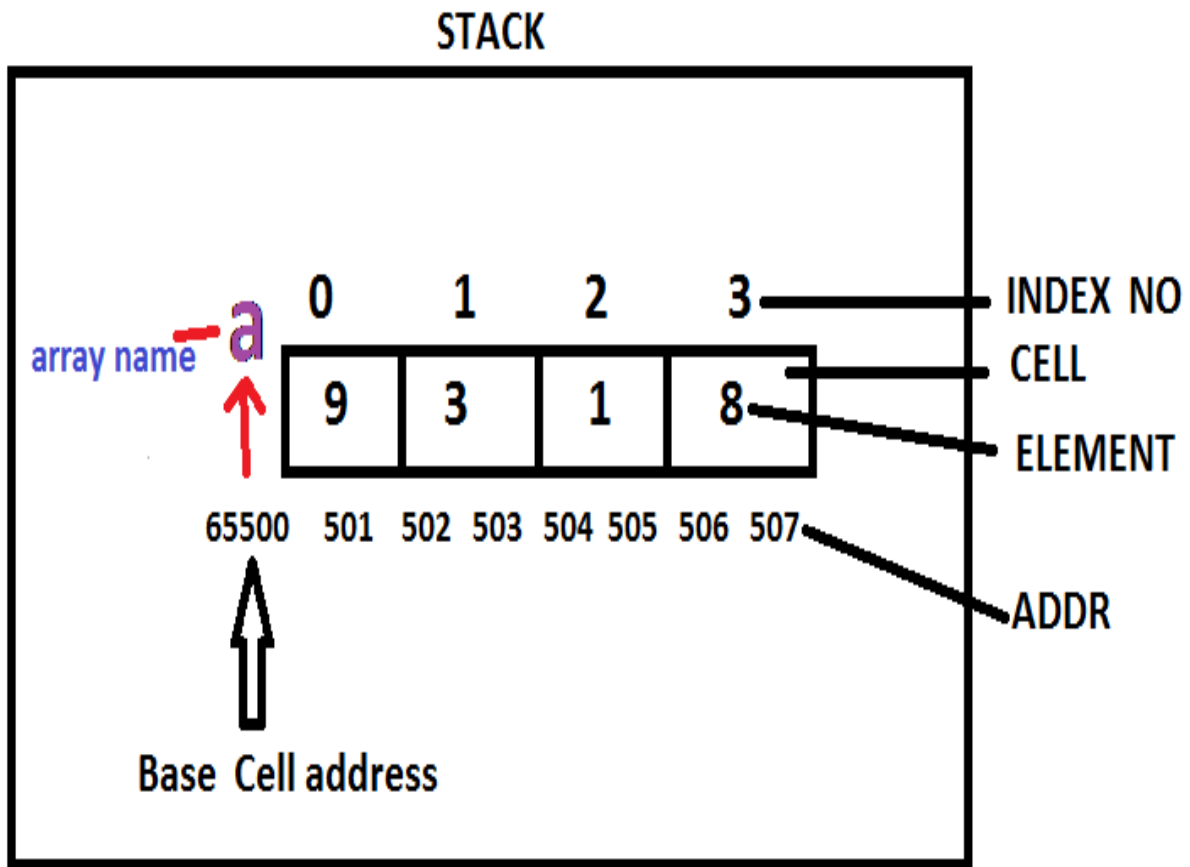
Syntax:

datatype variable[size] = {elements};

Eg:

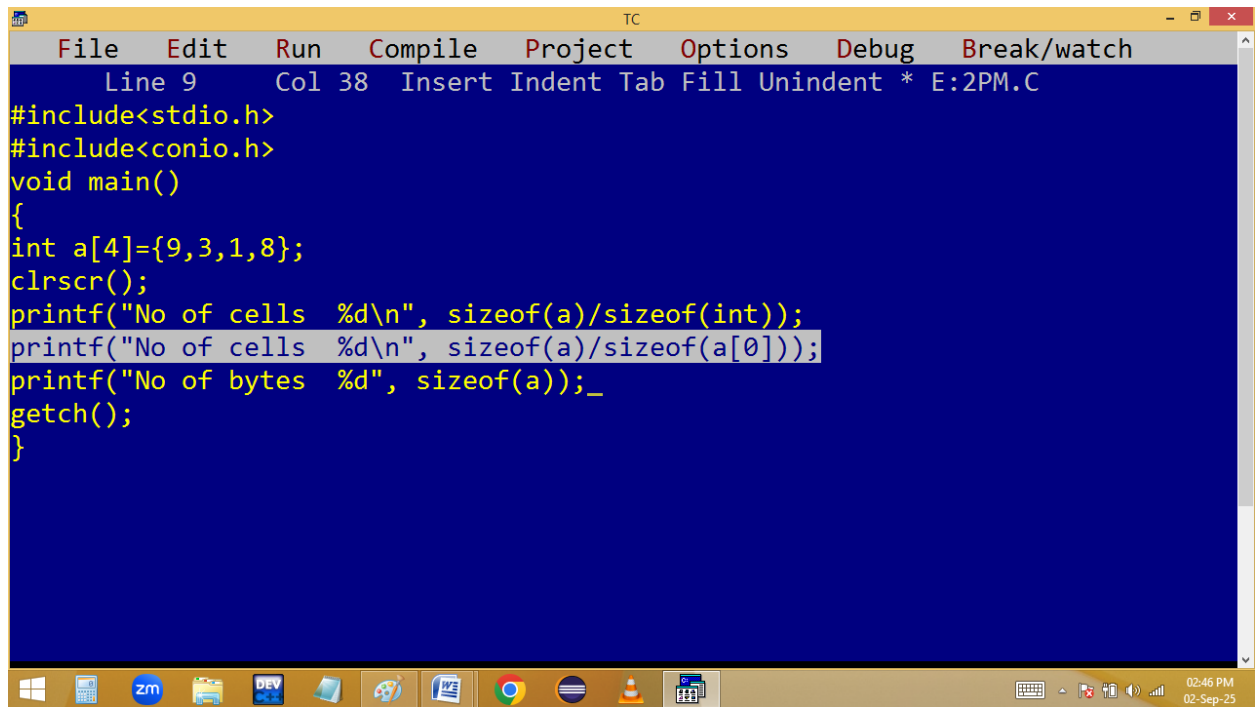
```
int a[4] = { 9, 3, 1, 8 };
```

Memory allocation for array:

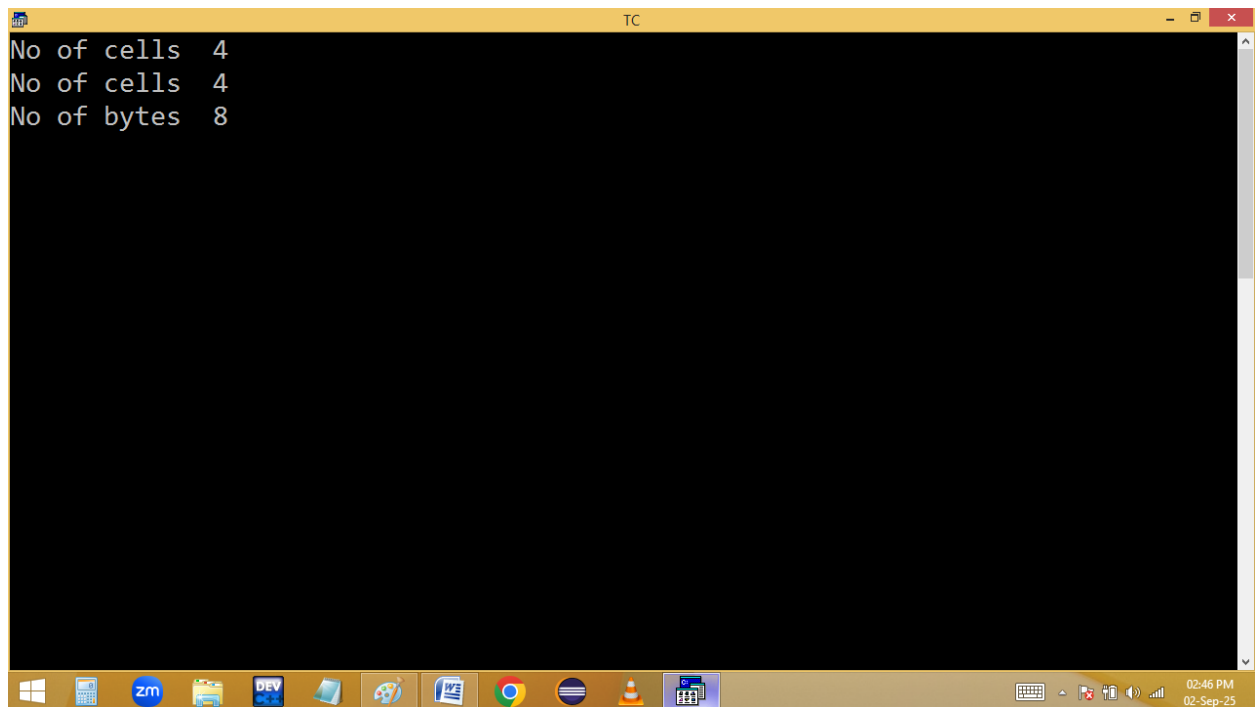


Array is implicit pointer because of array variable stores base cell [0 cell 1st byte] address. Hence array variable value and 0 cell address both are same.

Finding array size:

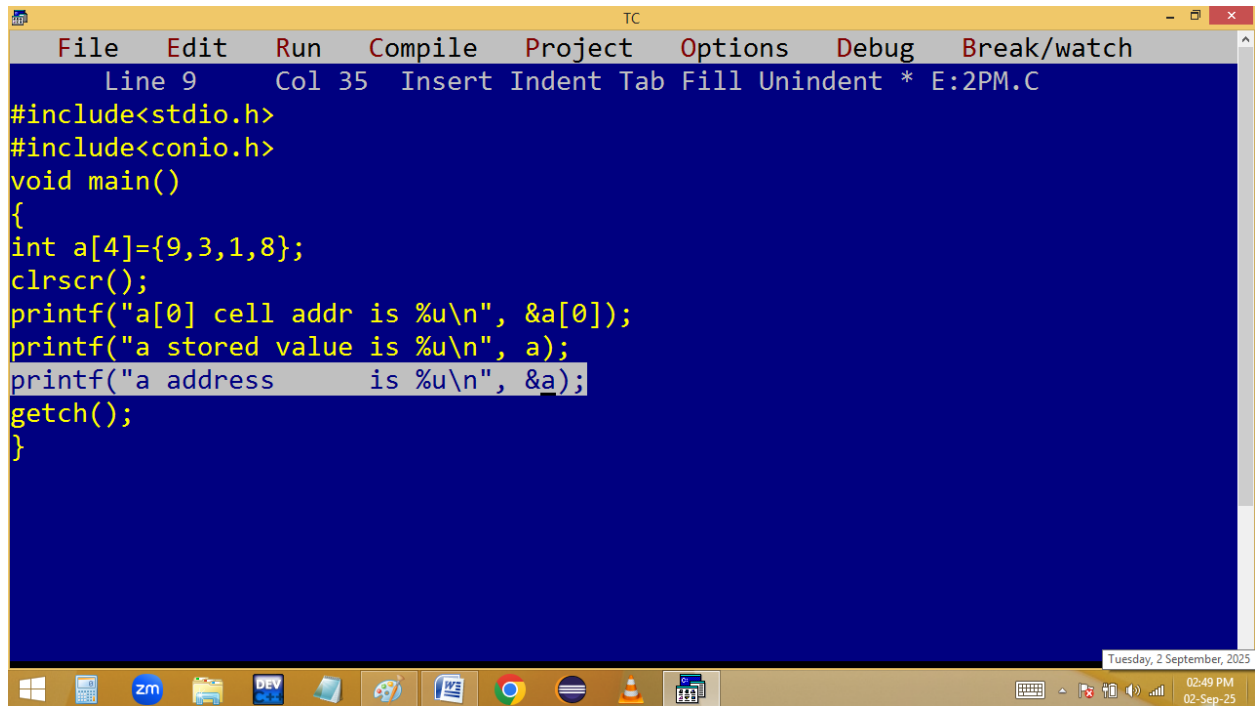


```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 38 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[4]={9,3,1,8};
clrscr();
printf("No of cells %d\n", sizeof(a)/sizeof(int));
printf("No of cells %d\n", sizeof(a)/sizeof(a[0]));
printf("No of bytes %d", sizeof(a));_
getch();
}
```



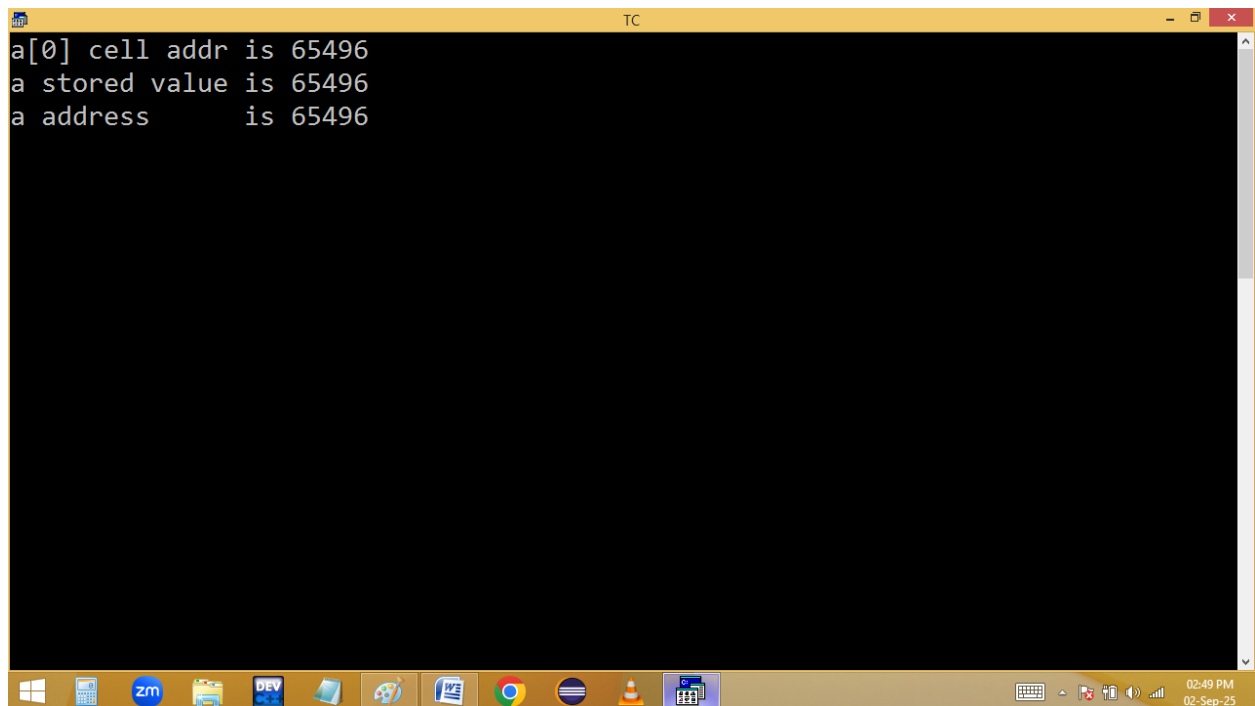
```
TC
No of cells 4
No of cells 4
No of bytes 8
```

Finding array address:



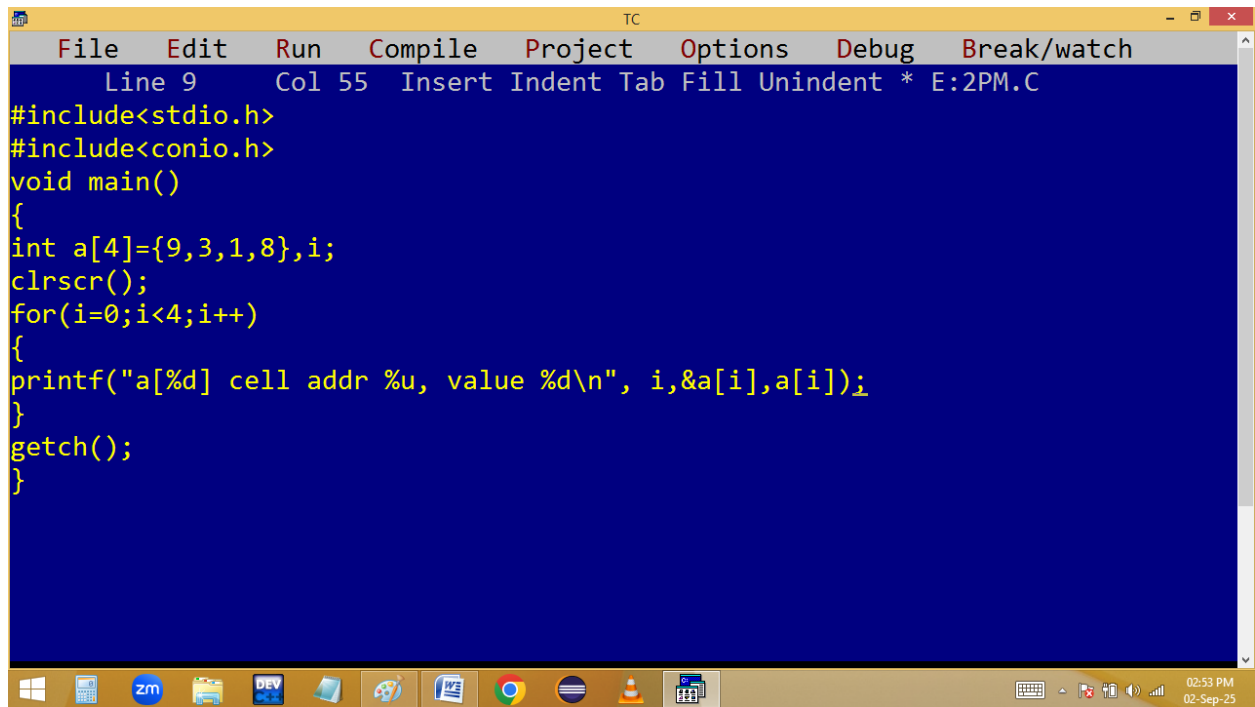
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 35 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[4]={9,3,1,8};
clrscr();
printf("a[0] cell addr is %u\n", &a[0]);
printf("a stored value is %u\n", a);
printf("a address is %u\n", &a);
getch();
}
```

Tuesday, 2 September, 2025 02:49 PM 02-Sep-25



```
TC
a[0] cell addr is 65496
a stored value is 65496
a address is 65496
```

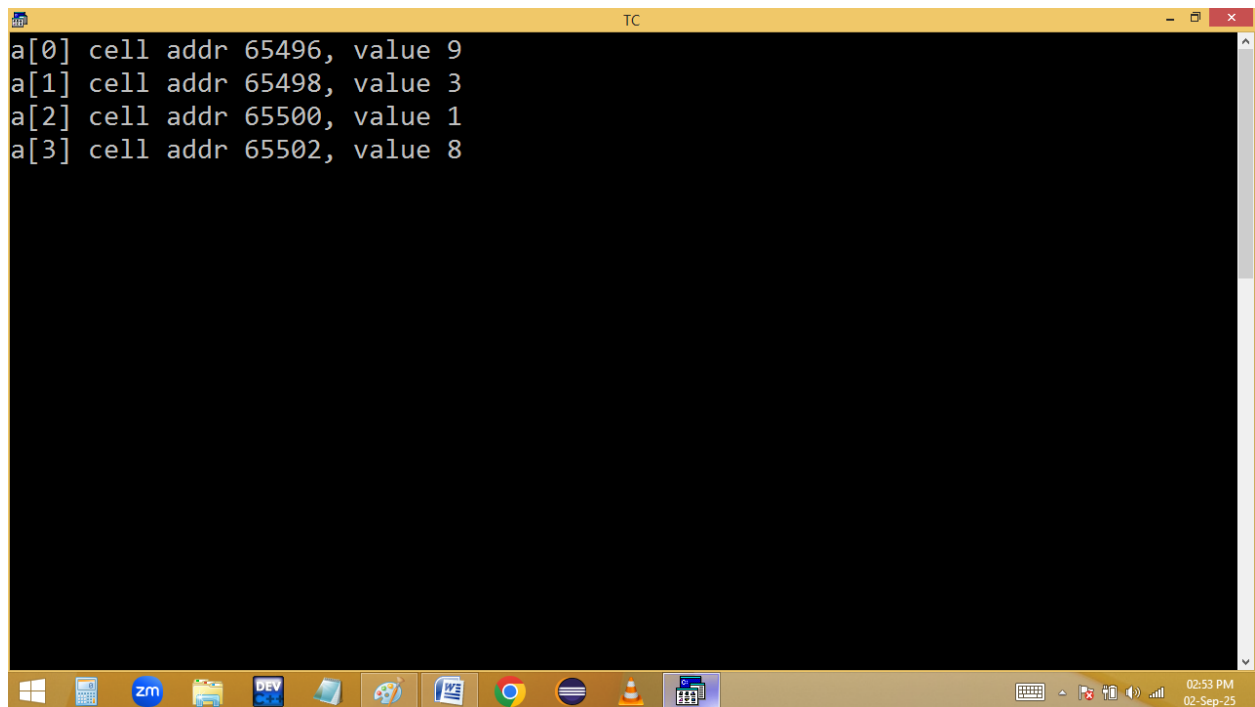
02:49 PM 02-Sep-25



The screenshot shows the Turbo C++ (TC) IDE with a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch) and a status bar (Line 9, Col 55, Insert, Indent, Tab, Fill, Unindent, * E:2PM.C). The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
void main()
{
int a[4]={9,3,1,8},i;
clrscr();
for(i=0;i<4;i++)
{
printf("a[%d] cell addr %u, value %d\n", i,&a[i],a[i]);
}
getch();
}
```

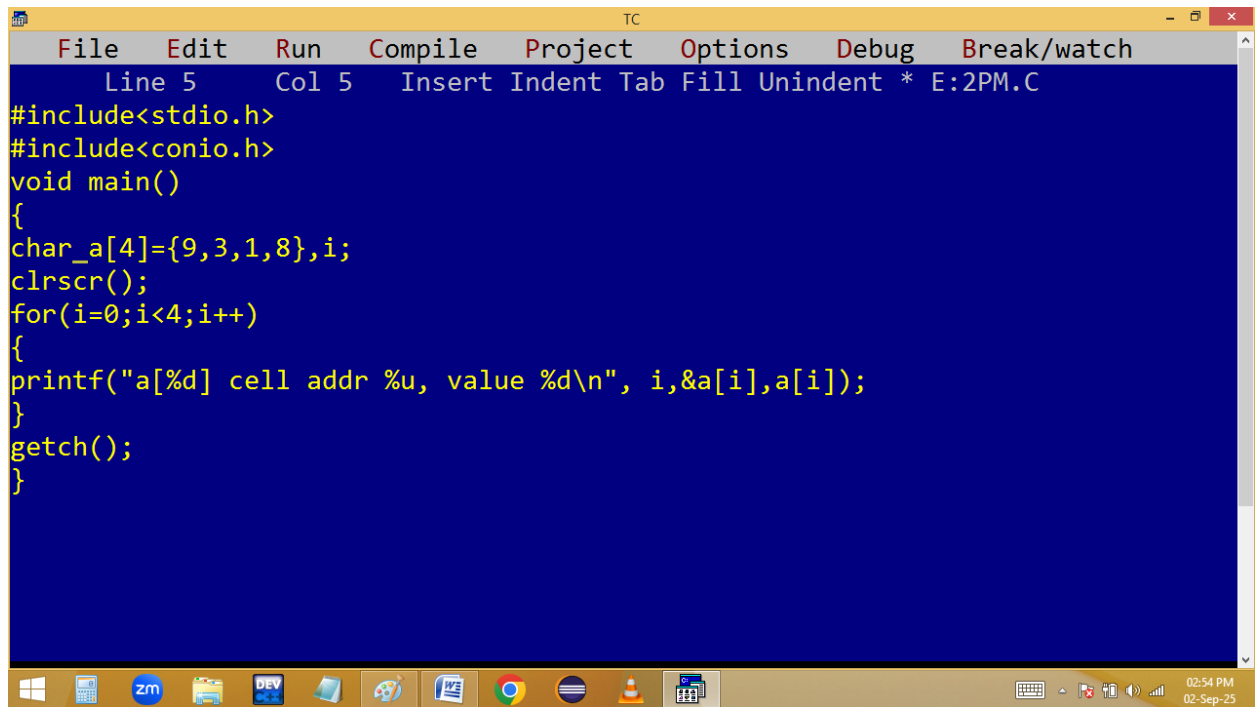
The Windows taskbar at the bottom shows various icons including Zoho Mail, DEV, and system tray icons for keyboard, volume, and network, with a timestamp of 02:53 PM on 02-Sep-25.



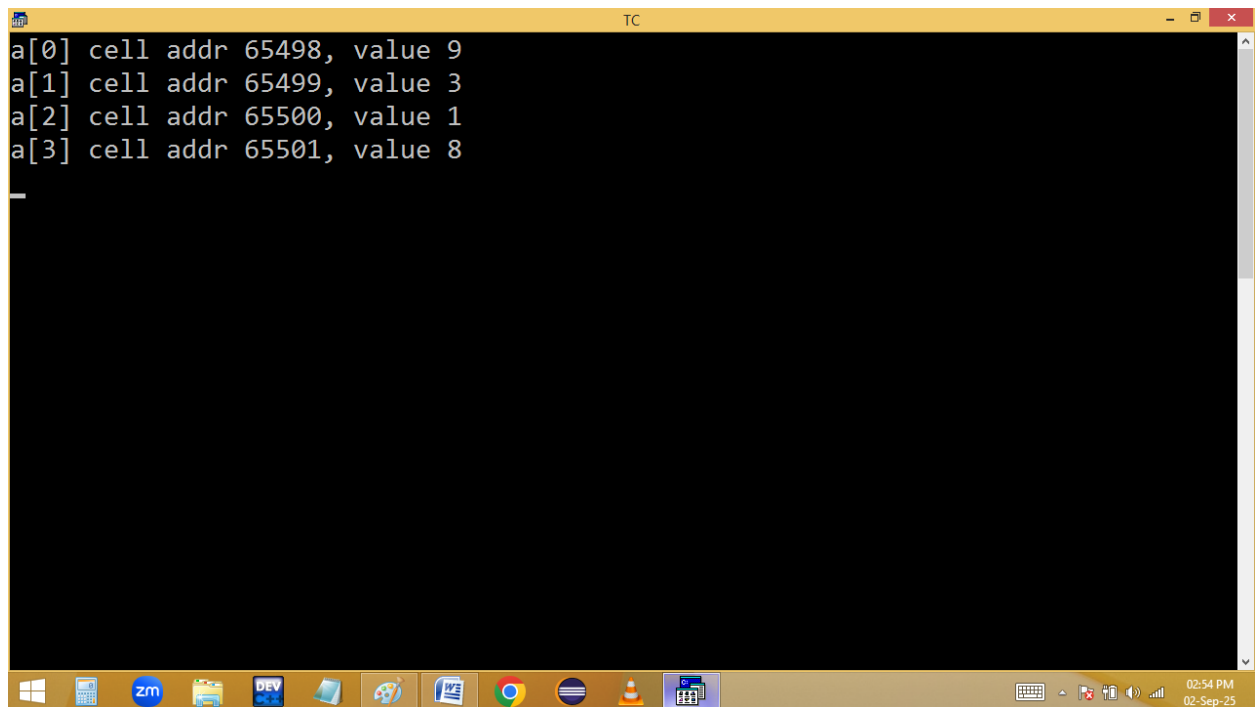
The screenshot shows the Turbo C++ (TC) IDE with the output of the program displayed in the console window. The output is as follows:

```
a[0] cell addr 65496, value 9
a[1] cell addr 65498, value 3
a[2] cell addr 65500, value 1
a[3] cell addr 65502, value 8
```

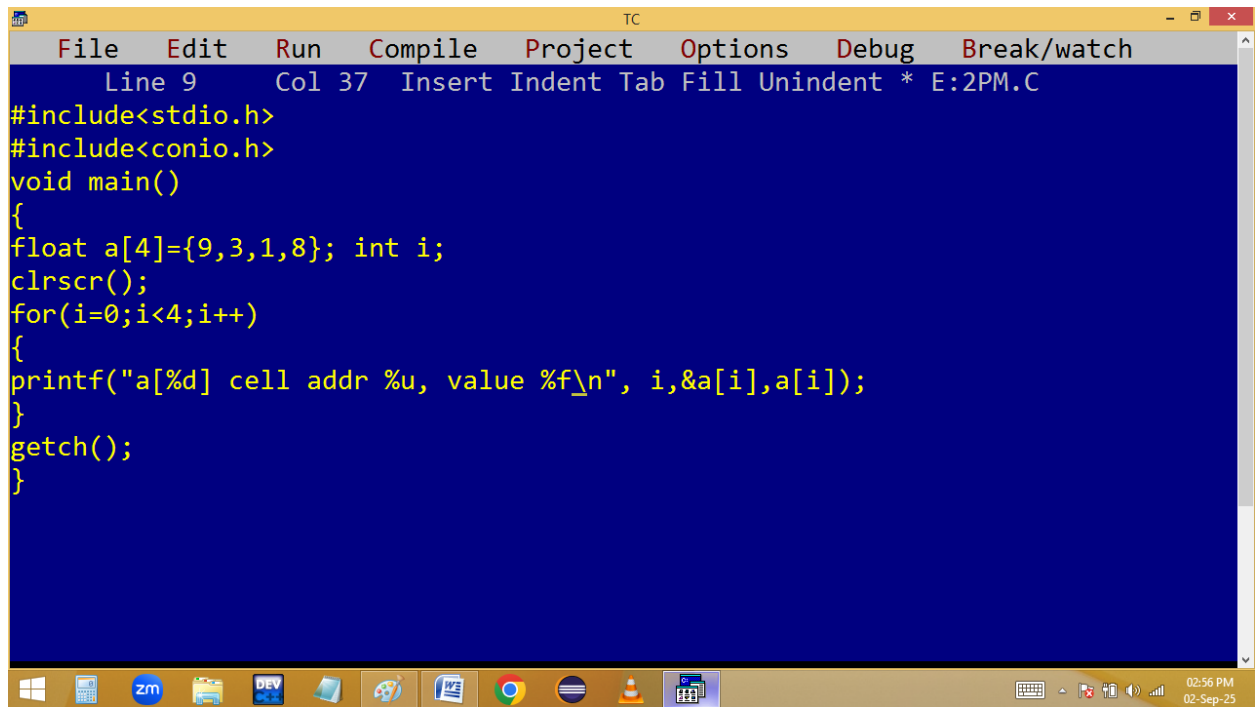
The Windows taskbar at the bottom is identical to the first screenshot, showing the same icons and system tray information (02:53 PM on 02-Sep-25).



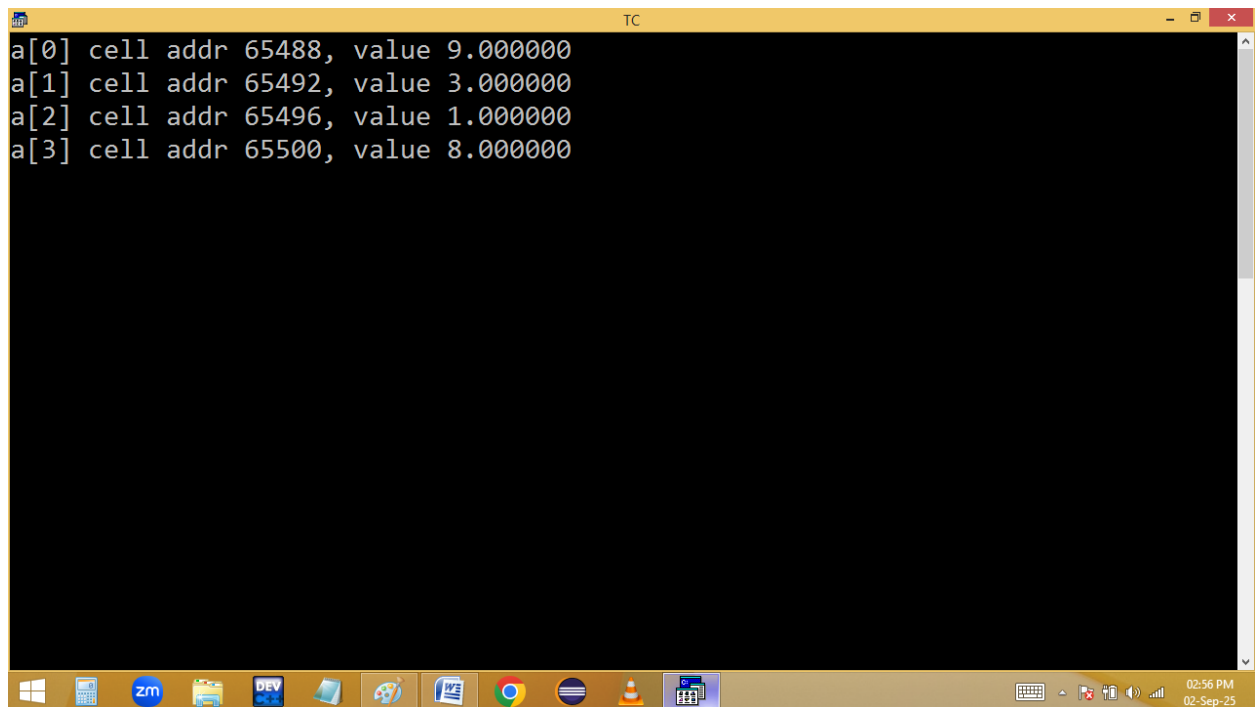
```
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 5 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
char_a[4]={9,3,1,8},i;
clrscr();
for(i=0;i<4;i++)
{
printf("a[%d] cell addr %u, value %d\n", i,&a[i],a[i]);
}
getch();
}
```



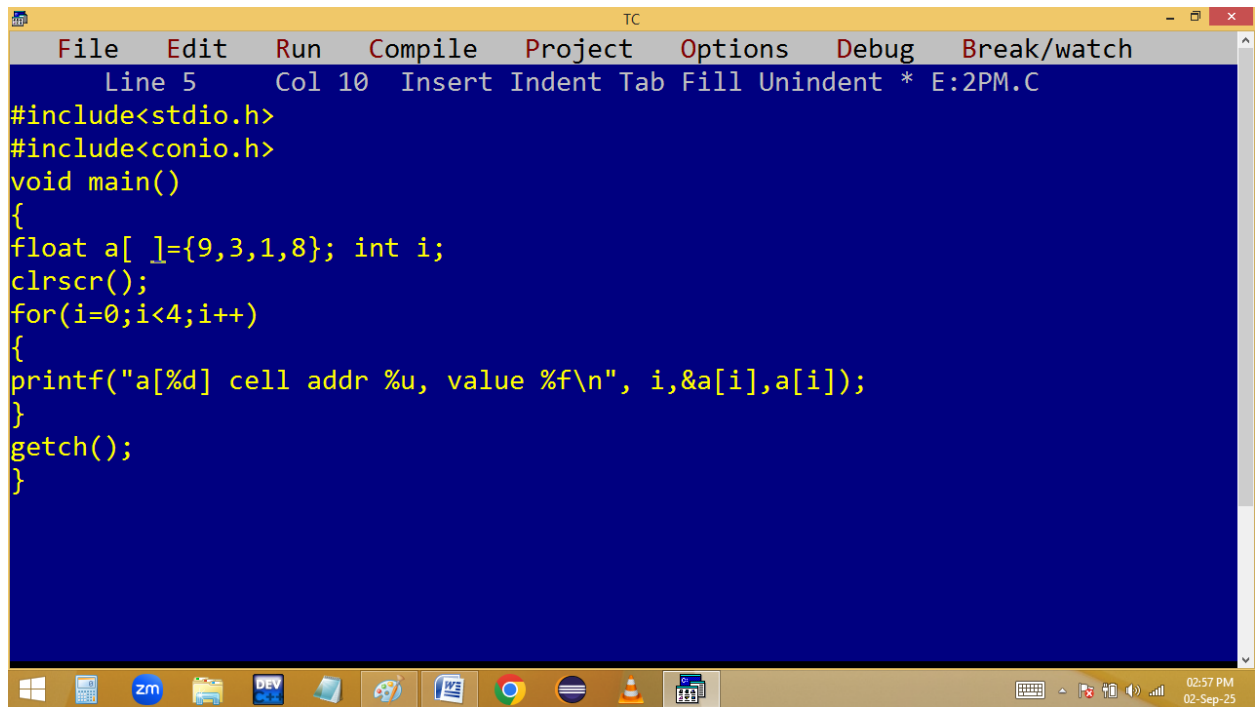
```
a[0] cell addr 65498, value 9
a[1] cell addr 65499, value 3
a[2] cell addr 65500, value 1
a[3] cell addr 65501, value 8
```



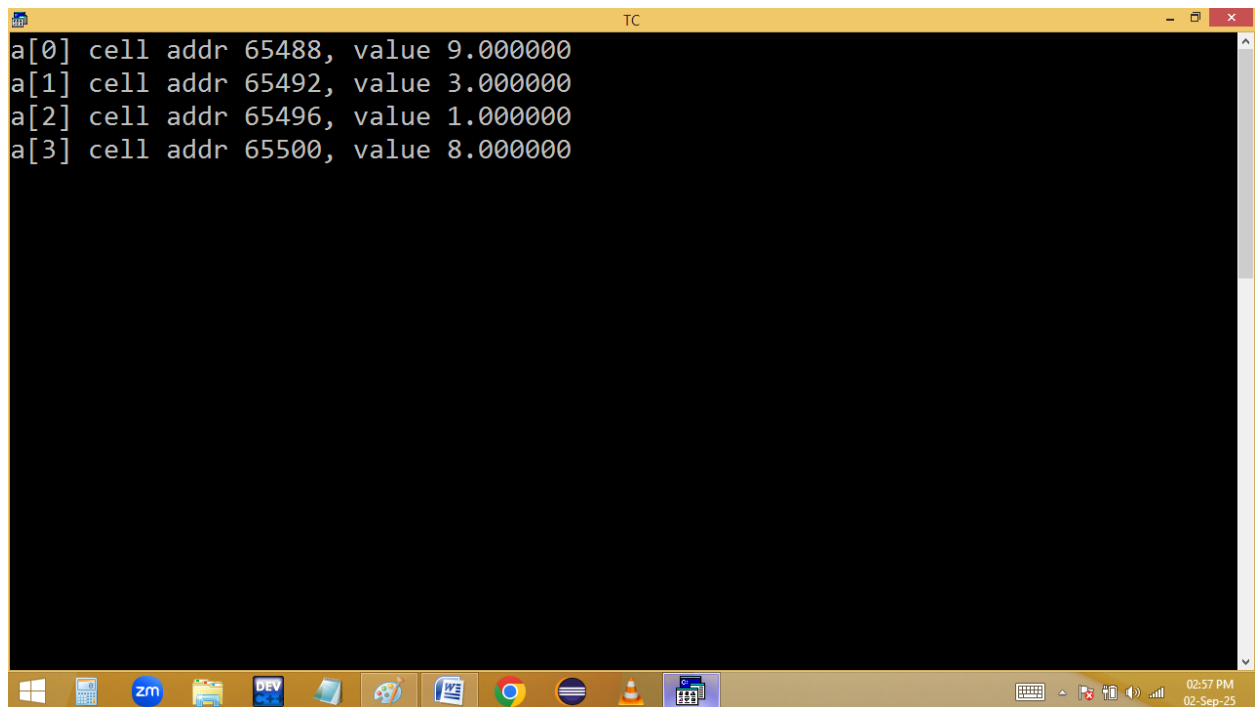
```
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 37 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
float a[4]={9,3,1,8}; int i;
clrscr();
for(i=0;i<4;i++)
{
printf("a[%d] cell addr %u, value %f\n", i,&a[i],a[i]);
}
getch();
}
```



```
a[0] cell addr 65488, value 9.000000
a[1] cell addr 65492, value 3.000000
a[2] cell addr 65496, value 1.000000
a[3] cell addr 65500, value 8.000000
```

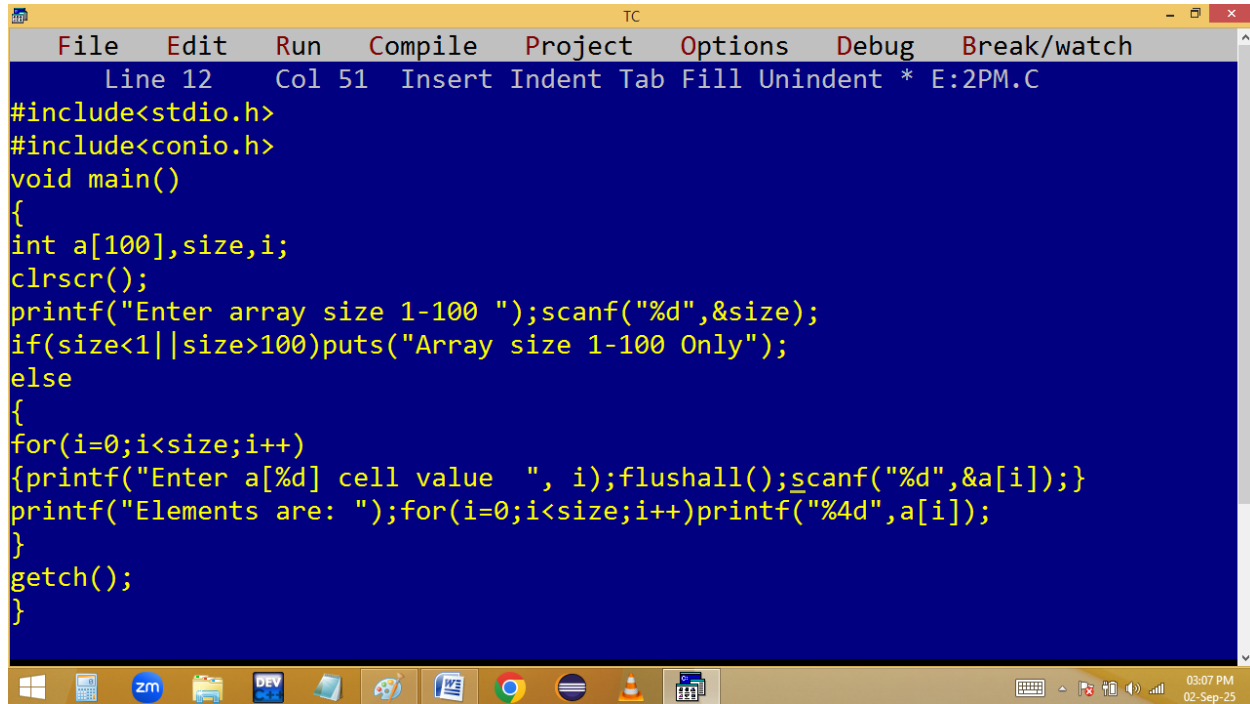



```
File Edit Run Compile Project Options Debug Break/watch
Line 5 Col 10 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
float a[ ]={9,3,1,8}; int i;
clrscr();
for(i=0;i<4;i++)
{
printf("a[%d] cell addr %u, value %f\n", i,&a[i],a[i]);
}
getch();
}
```

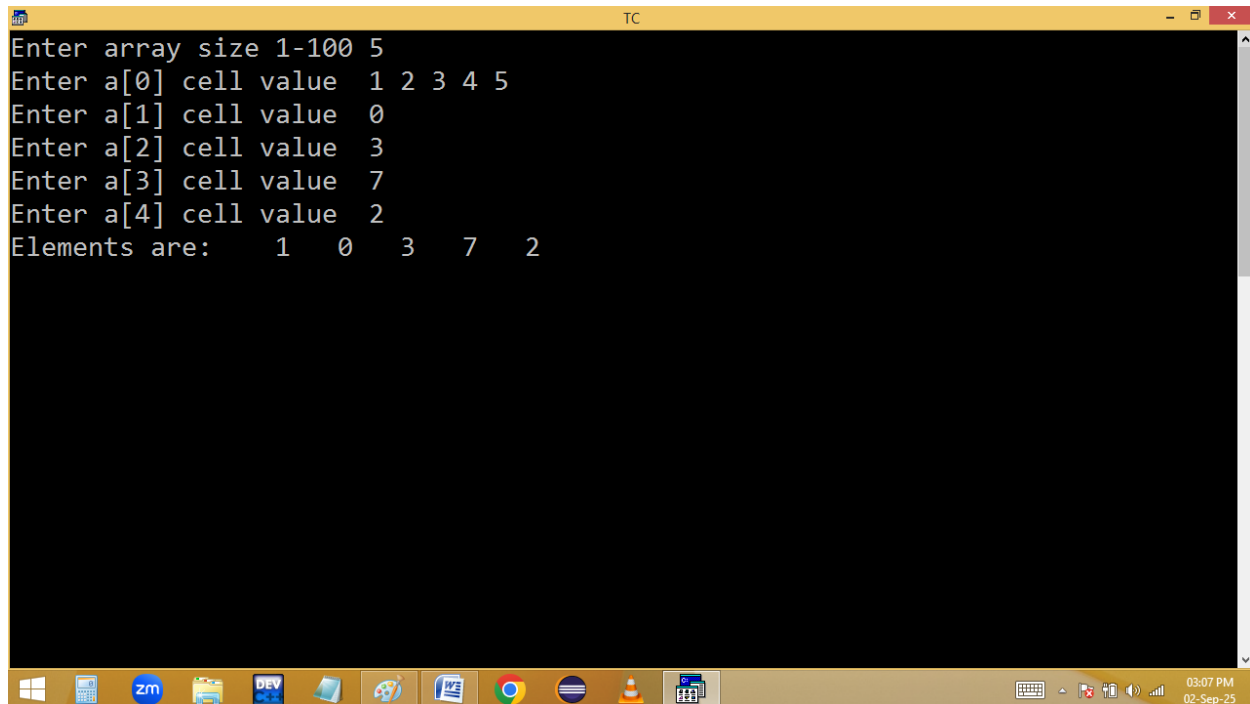


```
a[0] cell addr 65488, value 9.000000
a[1] cell addr 65492, value 3.000000
a[2] cell addr 65496, value 1.000000
a[3] cell addr 65500, value 8.000000
```

Reading and printing array elements at runtime:



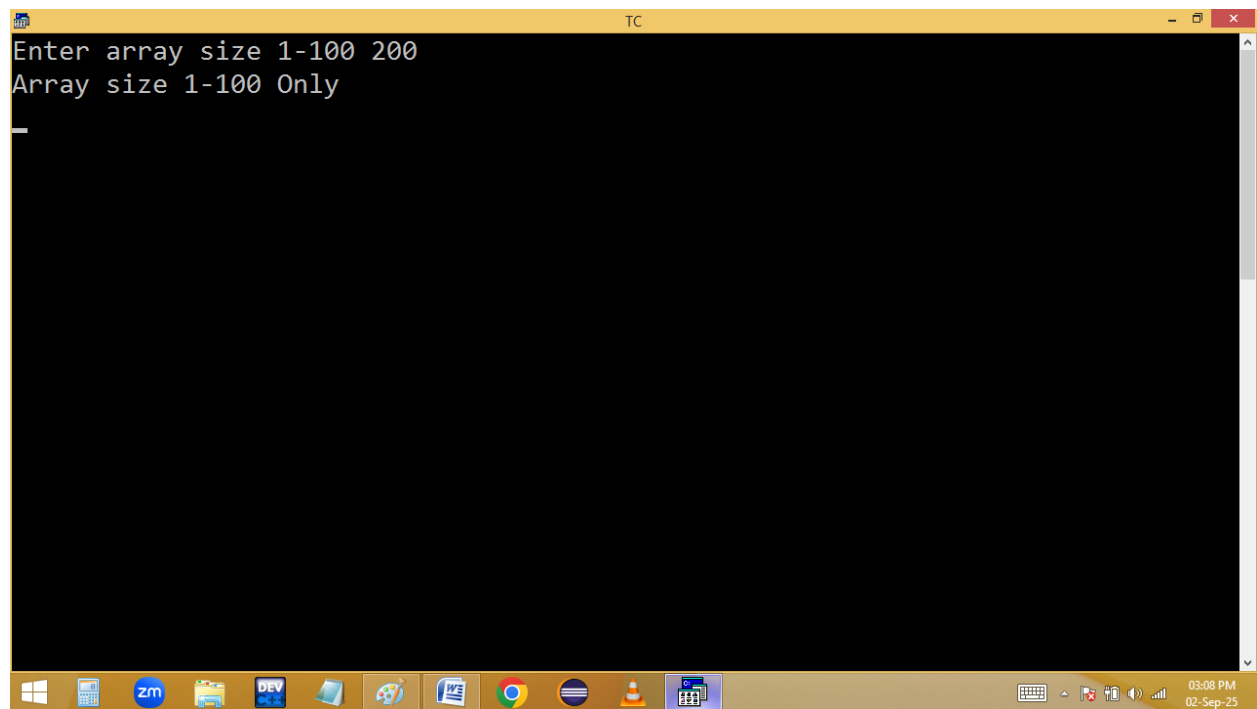
```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 12 Col 51 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],size,i;
clrscr();
printf("Enter array size 1-100 ");scanf("%d",&size);
if(size<1||size>100)puts("Array size 1-100 Only");
else
{
for(i=0;i<size;i++)
{printf("Enter a[%d] cell value ", i);flushall();scanf("%d",&a[i]);}
printf("Elements are: ");for(i=0;i<size;i++)printf("%4d",a[i]);
}
getch();
}
```



```
TC
Enter array size 1-100 5
Enter a[0] cell value 1 2 3 4 5
Enter a[1] cell value 0
Enter a[2] cell value 3
Enter a[3] cell value 7
Enter a[4] cell value 2
Elements are: 1 0 3 7 2
```

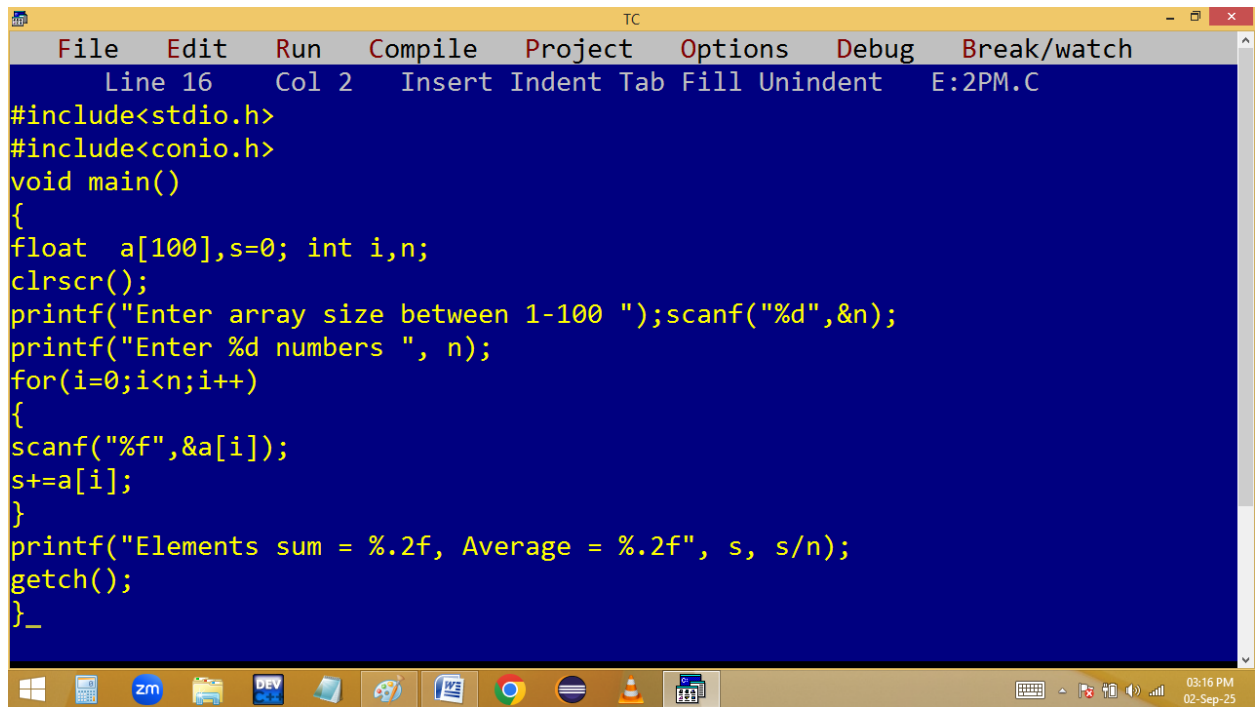
```
TC
Enter array size 1-100 3
Enter a[0] cell value 9
Enter a[1] cell value 0
Enter a[2] cell value 3
Elements are: 9 0 3_
```

```
TC
Enter array size 1-100 -5
Array size 1-100 Only
```

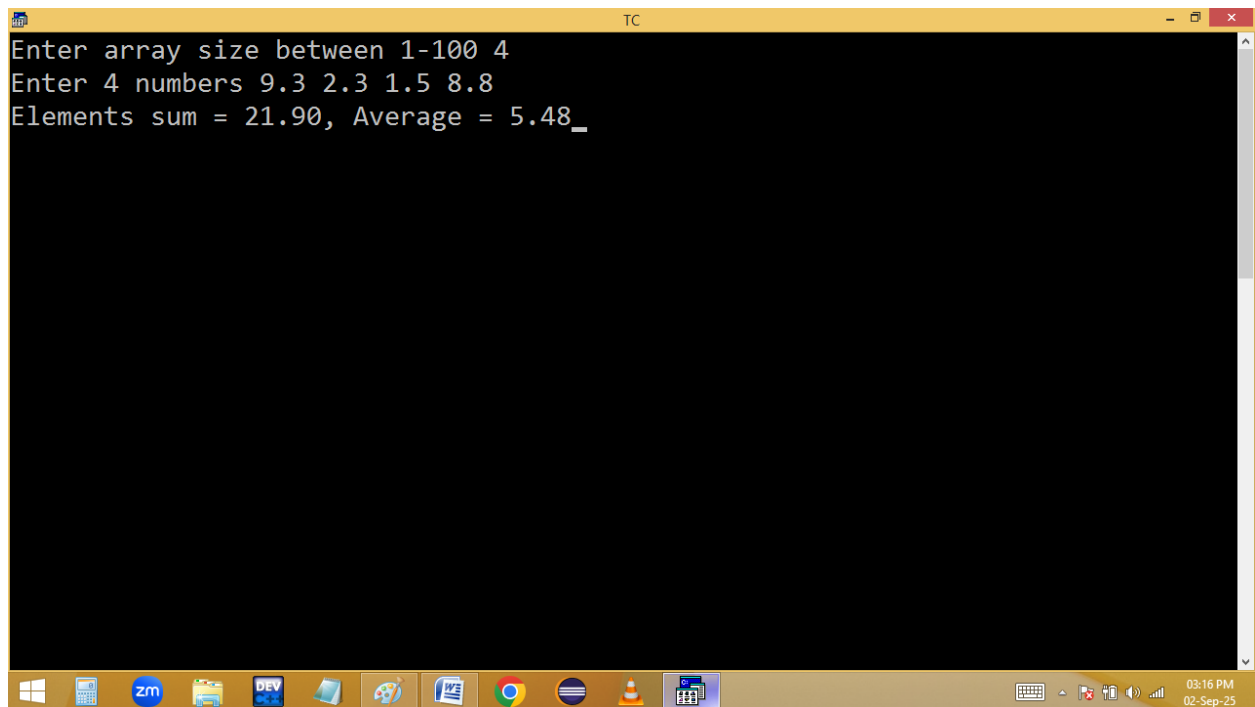


```
Enter array size 1-100 200
Array size 1-100 Only
```

Read n elements into array and find the elements sum and average?



```
File Edit Run Compile Project Options Debug Break/watch
Line 16 Col 2 Insert Indent Tab Fill Unindent E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
float a[100],s=0; int i,n;
clrscr();
printf("Enter array size between 1-100 ");scanf("%d",&n);
printf("Enter %d numbers ", n);
for(i=0;i<n;i++)
{
scanf("%f",&a[i]);
s+=a[i];
}
printf("Elements sum = %.2f, Average = %.2f", s, s/n);
getch();
}_
```



```
Enter array size between 1-100 4
Enter 4 numbers 9.3 2.3 1.5 8.8
Elements sum = 21.90, Average = 5.48_
```

```

Enter array size between 1-100 4
Enter 4 numbers 1 2 3 4
Elements sum = 10.00, Average = 2.50

```

```

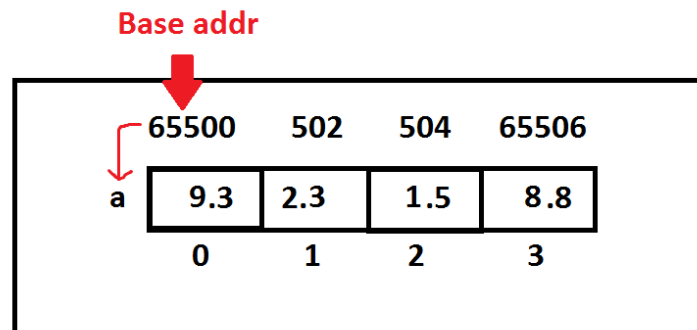
for( i=0; i<4; i++)
{
    scanf("%f", &a[i]);
    s+=a[i];
}

p("Sum=%.2f, Avg=%.2f", s, s/4);

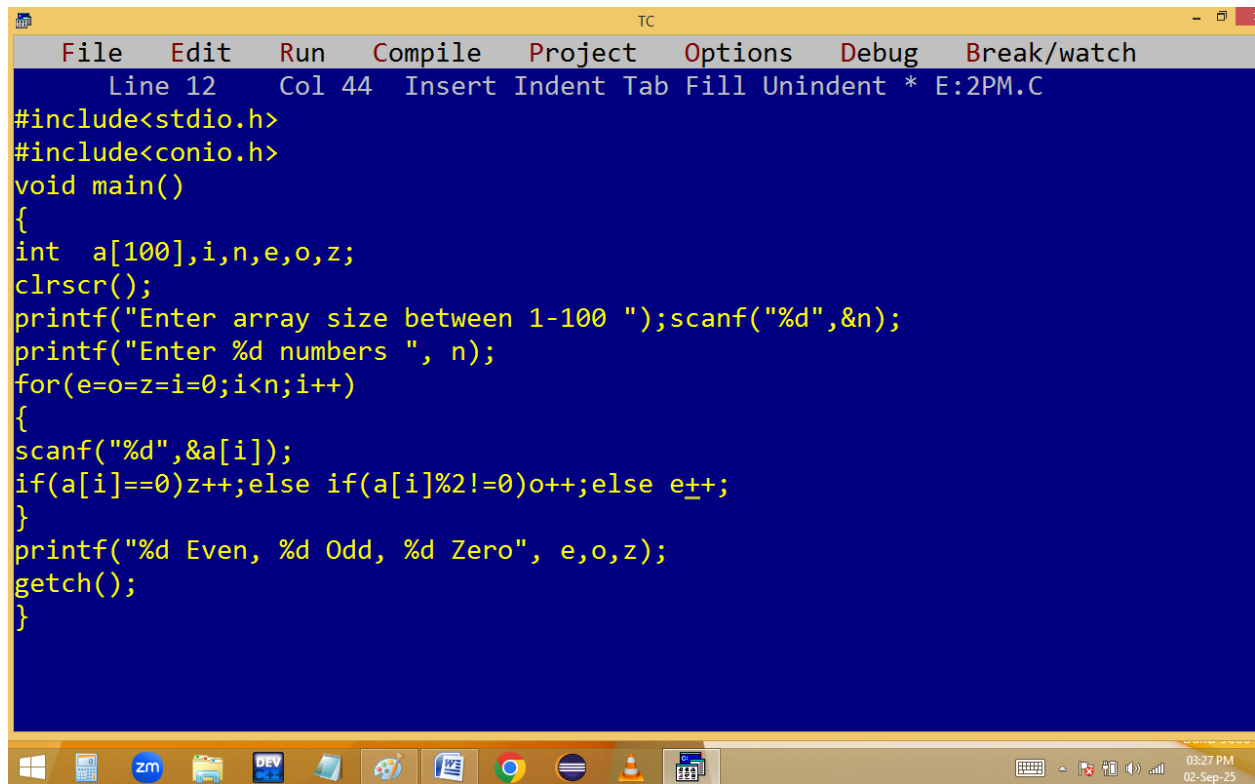
```

Handwritten calculations:

i	s
0	0
1	9.3
2	2.3
3	1.5
	8.8
	21.90



Finding no of even/odd/zero elements in array:



The image shows a screenshot of a Turbo C++ (TC) IDE. The window title is "TC". The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the top indicates "Line 12 Col 44 Insert Indent Tab Fill Unindent * E:2PM.C". 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  a[100],i,n,e,o,z;
clrscr();
printf("Enter array size between 1-100 ");scanf("%d",&n);
printf("Enter %d numbers ", n);
for(e=o=z=i=0;i<n;i++)
{
scanf("%d",&a[i]);
if(a[i]==0)z++;else if(a[i]%2!=0)o++;else e++;
}
printf("%d Even, %d Odd, %d Zero", e,o,z);
getch();
}
```

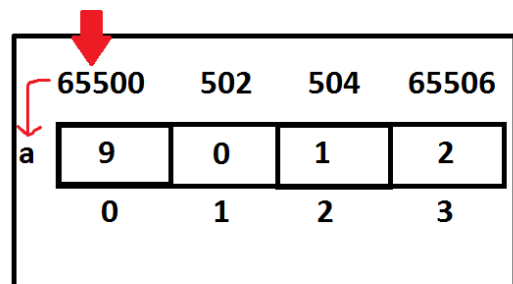
The Windows taskbar is visible at the bottom, showing icons for Windows, a calculator, Zoom, File Explorer, DEV C++, a folder, Paint, Notepad, Google Chrome, a globe icon, VLC media player, and a task manager icon. The system clock in the bottom right corner shows "03:27 PM 02-Sep-25".

```
TC
Enter array size between 1-100 4
Enter 4 numbers 1 2 3 0
1 Even, 2 Odd, 1 Zero
```

```
for(i=0;i<4;i++)
{
scanf("%d",&a[i]);
if(a[i]==0)z++; ✓
else if(a[i]%2==0)e++; ✓
else o++; ✓
}
p(e,o,z);
```

i	e	o	z
0	0	0	0
1	2	1	0
2	3	2	0
3			1

Base addr



65500	502	504	65506
9	0	1	2
0	1	2	3

Finding max, min elements of array:


```
TC
Line 16 Col 31 Insert Indent Tab Fill Unindent * E:2PM.C
#include<stdio.h>
#include<conio.h>
void main()
{
int a[100],i,max, min,n;
clrscr();
printf("Enter array size between 1-100 ");scanf("%d",&n);
printf("Enter %d numbers ", n);
for(i=0;i<n;i++)scanf("%d",&a[i]);
max=min=a[0];
for(i=1;i<n;i++)
{
if(max<a[i])max=a[i];
if(min>a[i])min=a[i];
}
printf("Max=%d, Min=%d",max, min);
getch();
}
```

Page: 16 of 17 Words: 304 100% 03:55 PM 02-Sep-25

```
TC
Enter array size between 1-100 10
Enter 10 numbers 3 0 1 7 33 -7 12 76 22 24
Max=76, Min=-7
```

Page: 17 of 17 Words: 304 100% 03:55 PM 02-Sep-25

$\frac{i}{1}$ $\frac{max}{9}$ $\frac{min}{9}$

```
for(i=1;i<4;i++)
{
    9 < 10
    if(max<a[i])max=a[i];
    if(min>a[i])min=a[i];
}
p( max, min );
```

Base addr

	65500	502	504	65506
a	9	10	1	-2
	0	1	2	3

Home work:

1. Arrange array elements in reverse order permanently.
2. Decimal to binary conversion

Eg: **20 ==> 10100**

