

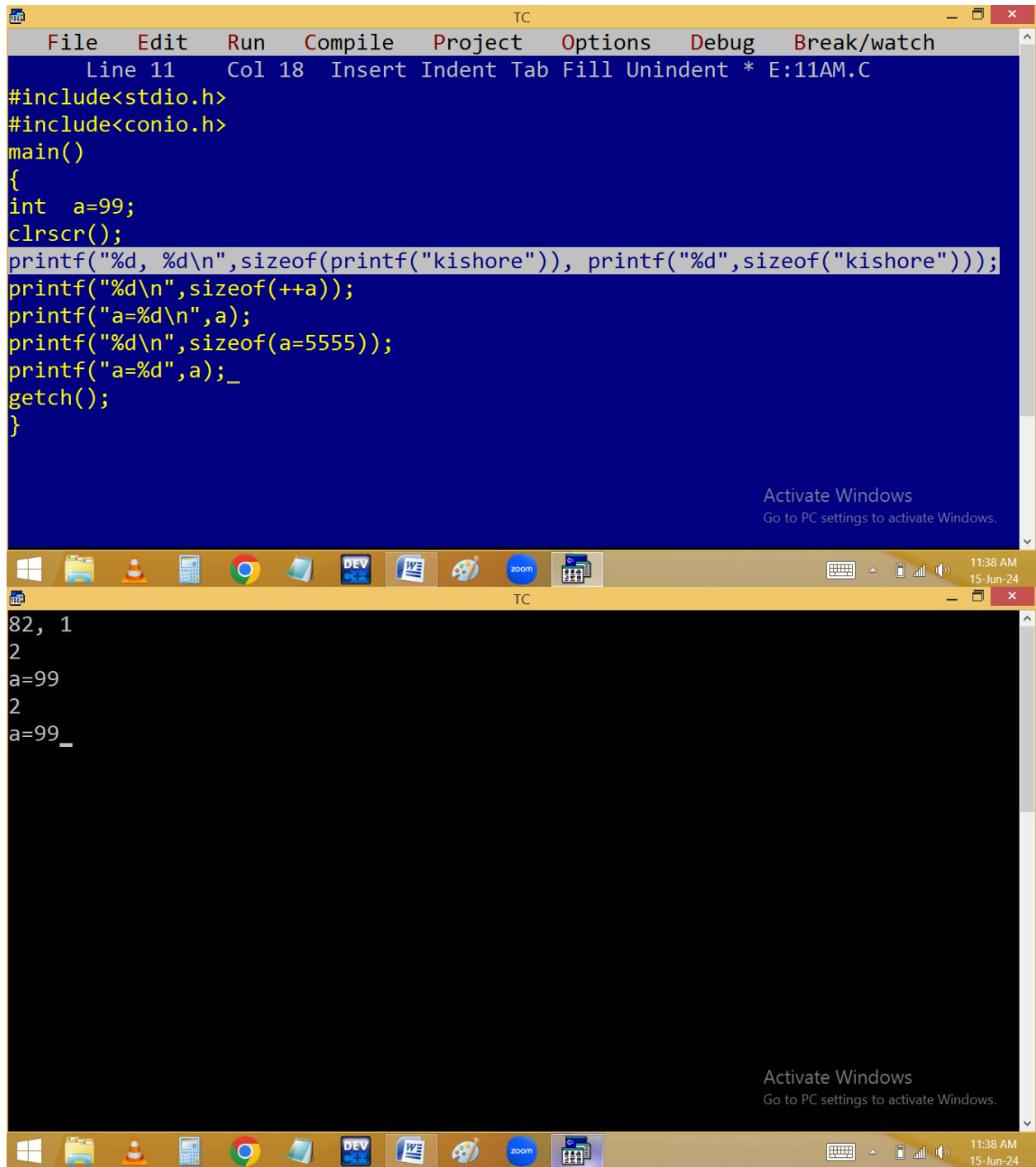
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 14 Col 64 Insert Indent Tab Fill Unindent * E:11AM.C
#include<stdio.h>
#include<conio.h>
main()
{
char s1[10]="Kishore", s2[]="Kishore";
clrscr();
printf("%d, %d\n",sizeof(s1), sizeof(s2));
printf("%d, %d\n",sizeof("kishore"), printf("kishore"));
printf("kishore addr is %u\n","kishore");
printf("%d, %d\n",sizeof("kishore")+1, sizeof("kishore")+1);
printf("%d, %d\n",sizeof("1.234"), sizeof("1.234")+10));
printf("%d, %d\n",sizeof(12,3.4), sizeof(3.4,12));
printf("%d, %d\n",sizeof(12+3.4), sizeof(3.4+12));
printf("%d, %d\n",sizeof(sizeof(1.8)), sizeof(sizeof("kishore")));
getch();
}
```

The bottom window shows the output of the program:

```
10, 8
kishore8, 7
kishore addr is 466
2, 9
6, 2
8, 2
8, 8
2, 2
```

The IDE interface includes a menu bar (File, Edit, Run, Compile, Project, Options, Debug, Break/watch), a status bar (Line 14, Col 64, Insert, Indent, Tab, Fill, Unindent, \* E:11AM.C), and a taskbar at the bottom with various application icons. The Windows taskbar at the very bottom shows the time as 11:30 AM on 15-Jun-24.



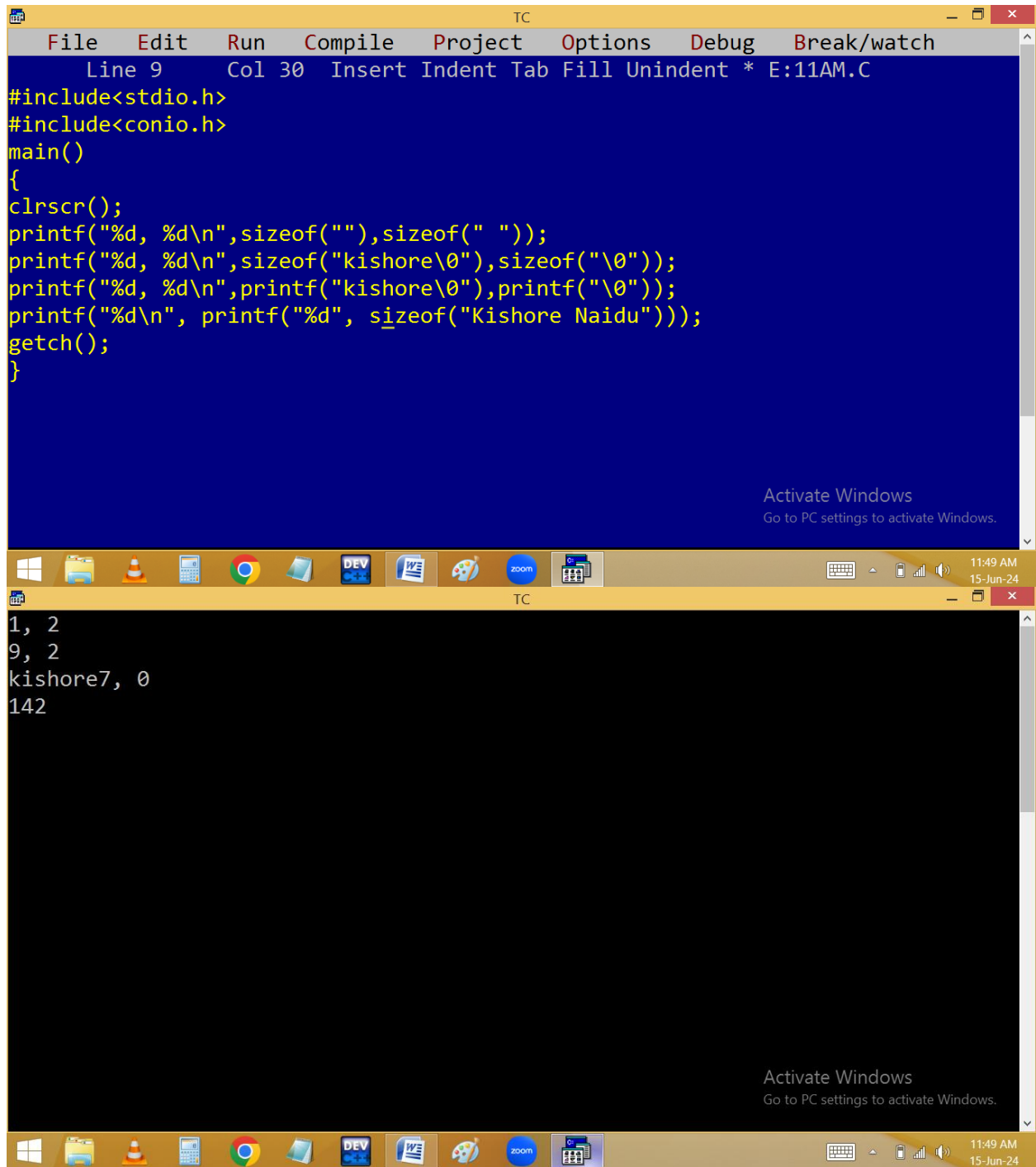
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window is the source code editor, which has a blue background and displays a C program. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. The status bar at the bottom of the editor shows 'Line 11 Col 18 Insert Indent Tab Fill Unindent \* E:11AM.C'. The code in the editor is as follows:

```
#include<stdio.h>
#include<conio.h>
main()
{
int a=99;
clrscr();
printf("%d, %d\n",sizeof(printf("kishore")), printf("%d",sizeof("kishore")));
printf("%d\n",sizeof(++a));
printf("a=%d\n",a);
printf("%d\n",sizeof(a=5555));
printf("a=%d",a);_
getch();
}
```

The bottom window is the output console, which has a black background. It displays the output of the program:

```
82, 1
2
a=99
2
a=99_
```

Both windows have a taskbar at the bottom with various application icons and a system tray showing the time as 11:38 AM on 15-Jun-24. An 'Activate Windows' watermark is visible in the bottom right corner of both windows.



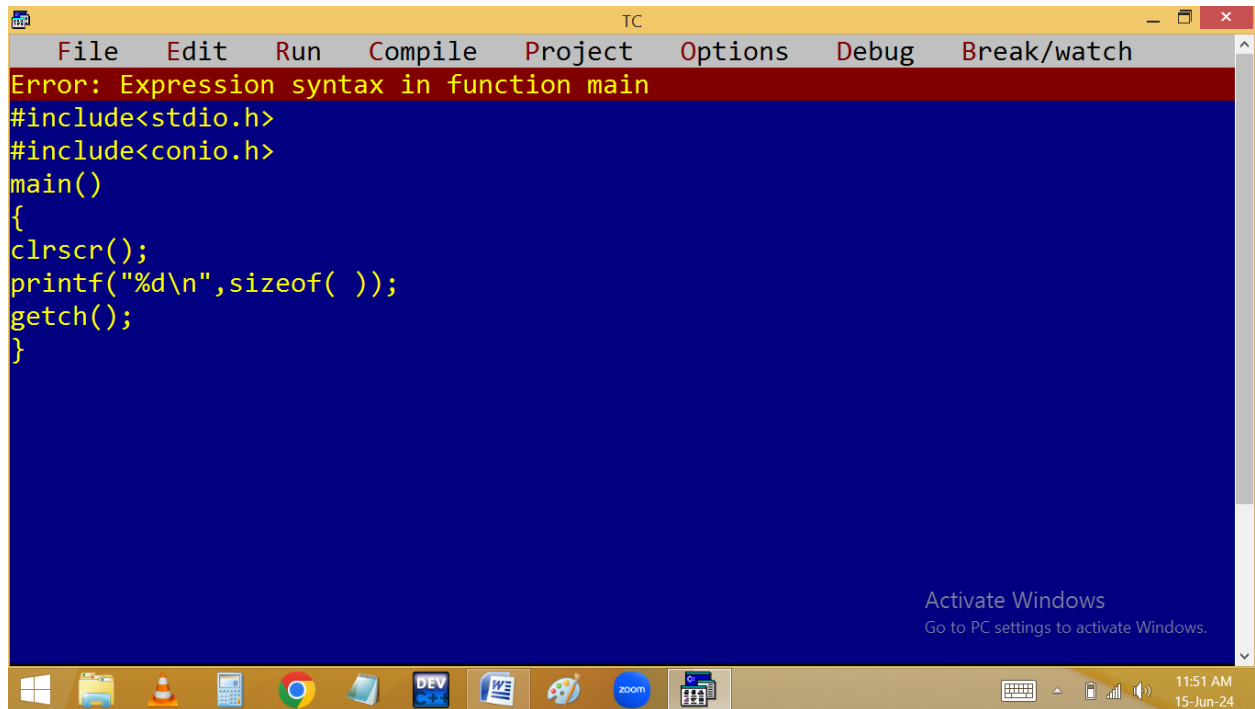
The image shows a screenshot of the Turbo C++ (TC) IDE. The top window displays a C program with the following code:

```
File Edit Run Compile Project Options Debug Break/watch
Line 9 Col 30 Insert Indent Tab Fill Unindent * E:11AM.C
#include<stdio.h>
#include<conio.h>
main()
{
clrscr();
printf("%d, %d\n",sizeof(""),sizeof(" "));
printf("%d, %d\n",sizeof("kishore\0"),sizeof("\0"));
printf("%d, %d\n",printf("kishore\0"),printf("\0"));
printf("%d\n", printf("%d", sizeof("Kishore Naidu")));
getch();
}
```

The bottom window shows the output of the program:

```
1, 2
9, 2
kishore7, 0
142
```

Both windows include a taskbar at the bottom with various application icons and a system tray showing the time as 11:49 AM on 15-Jun-24. An "Activate Windows" watermark is visible in the bottom right corner of both windows.



The screenshot shows the Turbo C++ (TC) IDE interface. The menu bar includes File, Edit, Run, Compile, Project, Options, Debug, and Break/watch. A red error message bar at the top states "Error: Expression syntax in function main". The code editor contains the following C code:

```
#include<stdio.h>
#include<conio.h>
main()
{
clrscr();
printf("%d\n",sizeof( ));
getch();
}
```

The Windows taskbar at the bottom shows various icons including File Explorer, VLC, Chrome, and the system clock indicating 11:51 AM on 15-Jun-24.

## Address operators:

1. & - Address of the variable
2. \* - pointer [ addr of another variable ]

```
TC
File Edit Run Compile Project Options Debug Break/watch
Line 11 Col 34 Insert Indent Tab Fill Unindent * E:11AM.C
#include<stdio.h>
#include<conio.h>
main()
{
int a=100;
int *p; /* ptr var */
clrscr();
p = &a; /* p stores a addr */
printf("a value is %d\n",a);
printf("a addr is %u\n",&a);
printf("p stored value is %u",p);_
getch();
}

a value is 100
a addr is 65502
p stored value is 65502_

TC
a value is 100
a addr is 65502
p stored value is 65502_

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

In devc++:

```
#include<stdio.h>
```

```
#include<conio.h>
```

```

main()
{
int a=1,b;

b = a++ + a++ + a++;

printf("a=%d, b=%d\n",a,b);

a=1;

b = ++a + ++a + ++a;

printf("a=%d, b=%d\n",a,b);

getch();
}

```

$a=1$   
 $\nearrow 2 \nearrow 3 \nearrow 4$   
 $b = \underline{a}++ + \underline{a}++ + \underline{a}++;$   
 $\uparrow \quad \uparrow \quad \uparrow$   
 $1 + 2 + 3 = 6$

$a=1$  ~~2~~ 3 4 ✓  
 $b = ++a + ++a + ++a;$   
 $\nearrow \nearrow \nearrow$   
 $3 + 3 + 4 = 10$  ✓

## BITWISE OPERATORS

Bitwise operator's works on bits.

Turbo-c is a 16 bit compiler. Due to this bitwise operations are limited to 16 bits only [ $2^0$  to  $2^{15}$ ].

Bitwise operators operate **integer** type values only.

We have to calculate only the **on** bits [ **1** ].

When the first bit[**Sign bit**] is **1** then the number is **Negative** and it is **0** then the number is **positive**.

They are very much used in system software development.

**Note:** Bitwise operator is low level feature.

C-Language supports following bitwise operators.

**&** -Bitwise and

**|** - Bitwise or

**^** - XOR ==> Exclusive OR

**~** - Compliment operator

**<<** - Left shift operator

**>>** - Right shift operator

**& - Bitwise and:** In this both bits are 1's then result bit is 1. Otherwise result bit is 0.

Eg: **25 & 15 = 9**

25 = 0000 0000 0001 1001  
15 = 0000 0000 0000 1111

2 | 25  
2 | 12 - 1  
2 | 6 - 0  
2 | 3 - 0  
1 - 1

2 | 15  
2 | 7 - 1  
2 | 3 - 1  
1 - 1

**25 & 15 = 9**

25 = 0000 0000 0001 1001  
15 = 0000 0000 0000 1111

&

0000 0000 0000 1001

↓ ↓  
 $2^3 + 2^0$   
↓ ↓  
**8 + 1 = 9**



**| - Bitwise or:** In this both bits are 0's then result bit is 0. Otherwise result bit is 1.

Eg:  $25 \mid 15 = 31$

**25 | 15 = 31**

25 = 0000 0000 0001 1001

15 = 0000 0000 0000 1111

$$\begin{array}{r} 0000 \ 0000 \ 0001 \ 1001 \\ 0000 \ 0000 \ 0000 \ 1111 \\ \hline 0000 \ 0000 \ 0001 \ 1111 \end{array}$$

$2^4 + 2^3 + 2^2 + 2^1 + 2^0$

$16 + 8 + 4 + 2 + 1 = 31$

**^ - XOR [Exclusive OR]:** In this both bits are same then result bit is 0. Otherwise result bit is 1.

Eg:  $25 \wedge 15 = 22$

$$25 \wedge 15 = 22$$

25 = 0000 0000 0001 1001

15 = 0000 0000 0000 1111

$\wedge$

---

0000 0000 0001 0110

$$\begin{array}{c} 2^4 + 2^2 + 2^1 \\ 16 + 4 + 2 = 22 \end{array}$$

**~ - Complement operator:** In complement operation the bits are complimented. i.e. 1's become 0's and 0's become 1's. Due to this +Ve no becomes -Ve and -Ve no becomes +Ve.

eg: ~25 → -26

25 = 0000 0000 0001 1001  
1111 1111 1110 0110  
-128+64+32+4+2=-26  
-128 + 102 = -26

25 = 0000 0000 0001 1001

~ = 1111 1111 1110 0110

5 2 1

$$2+4+32+64+128+256+512+1024+2048+4096+8192+16384-32768=-26$$

~-25 = 0000 0000 0001 1001

1's ~ = 1111 1111 1110 0110

2's ~ = 0000 0000 0000 0001  
1111 1111 1110 0111

24 2  
16+8=24

|   |   |   |    |
|---|---|---|----|
| 1 | 0 | 0 | 01 |
| 0 | 1 | 0 | 1  |
| 1 | 1 | 0 | 10 |

**Note: When starting bit is 1 given no is –  
Ve.**

**Eg: ~-25 → +24**

~-25 = +24

25 = 0000 0000 0001 1001  
1111 1111 1110 0110

<== 1's compliment

+1

<== 2's Compliment

1111 1111 1110 0111  
0000 0000 0001 1000

↓ ↓

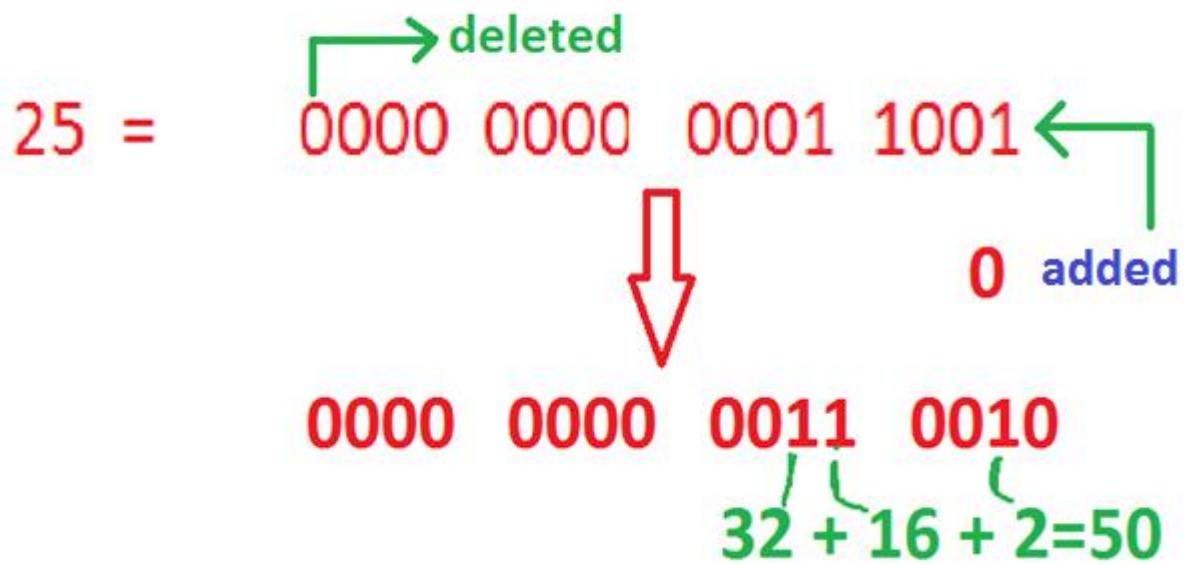
16+8=24

## **<< - left shift operator:**

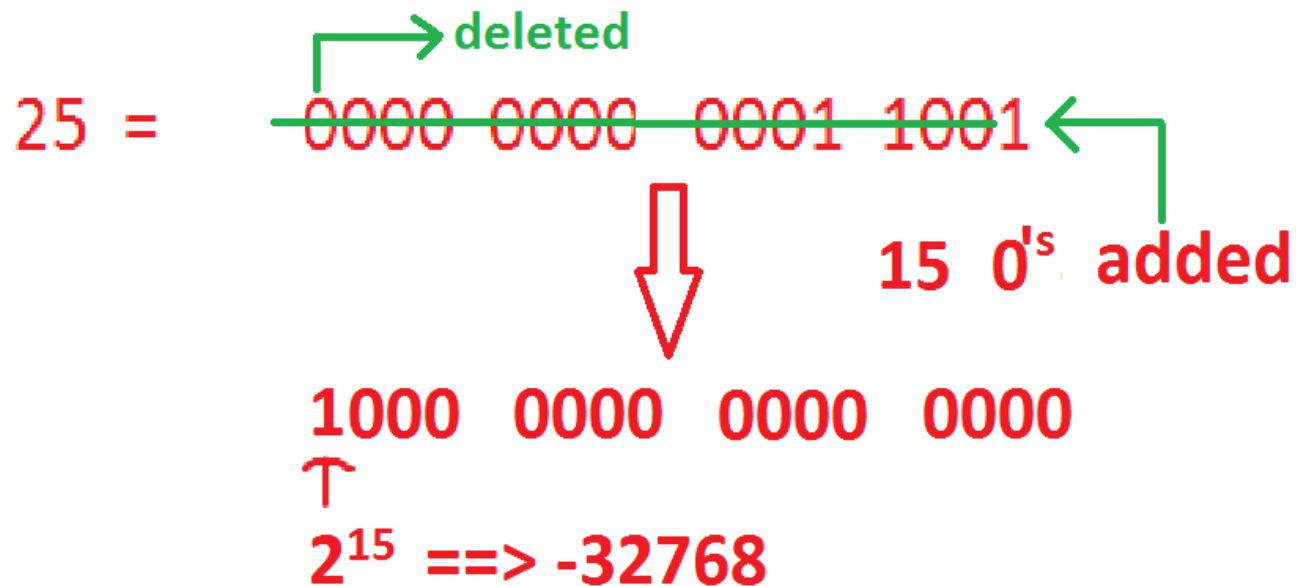
In left shift operation, the specified no of bits are deleted from left side and the same no of **zeros** added on right side. In left shift operation, most probably the value is multiplied with 2 that no of times.

**Eg:**  $25 \ll 1 = 50$ ,  $25 \ll 2 = 100$ ,  $25 \ll 15 = -32768$ ,  
 $25 \ll 16 = 0$

eg:  $25 \ll 1 = 50$



eg:  $25 \ll 15 = -32768$



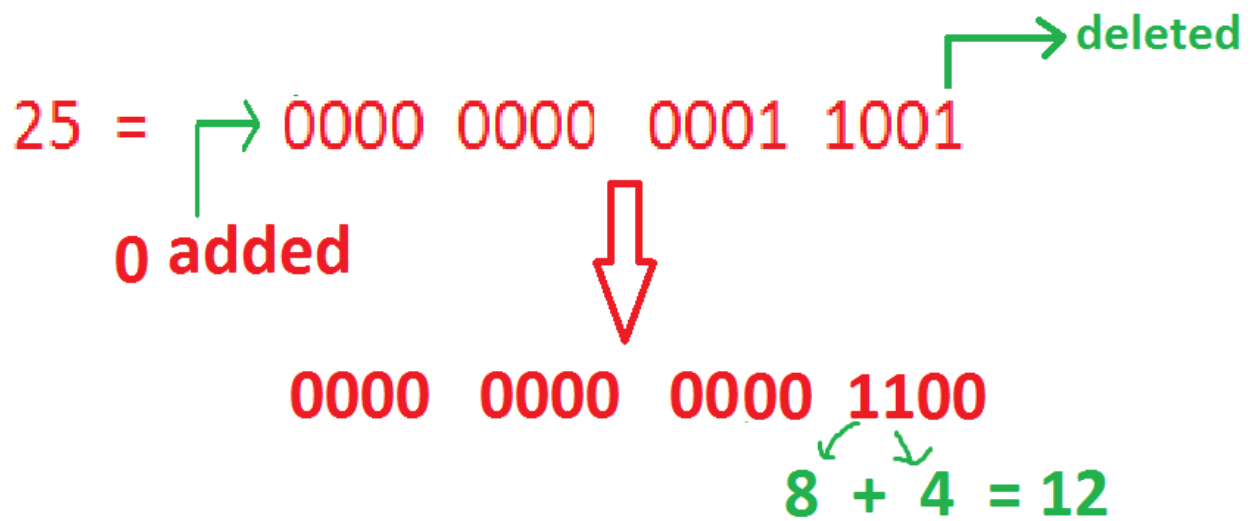
Note: When starting bit 1 no is negative.

### >> - Right shift operator:

In right shift operation, the bits are moved to right side i.e. the specified no.of bits are deleted from right side and same no.of **zero's** are added left side. Due to this always the number is divided with 2 that no of times.

Eg:  $25 \gg 1 = 12$ ,  $25 \gg 2 = 6$ ,  $25 \gg 3 = 3$ ,  $25 \gg 4 = 1$ ,  $25 \gg 5 = 0$

eg:  $25 \gg 1 = 12$



eg:  $25 \gg 5 = 0$

