INDIAN INSTITUTE OF TECHNOLOGY ROORKEE

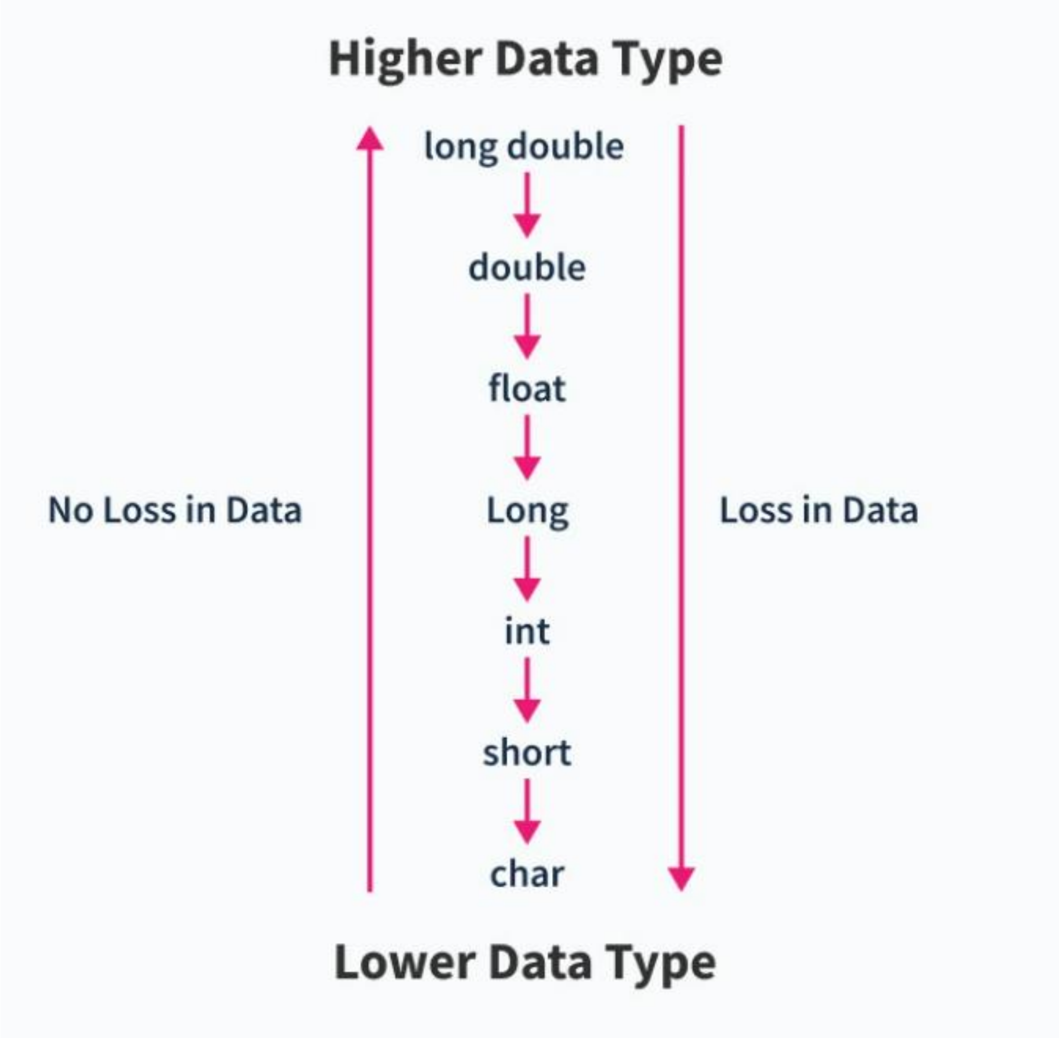
# Programming with C and C++

*CSC-101 (Lecture 08)*

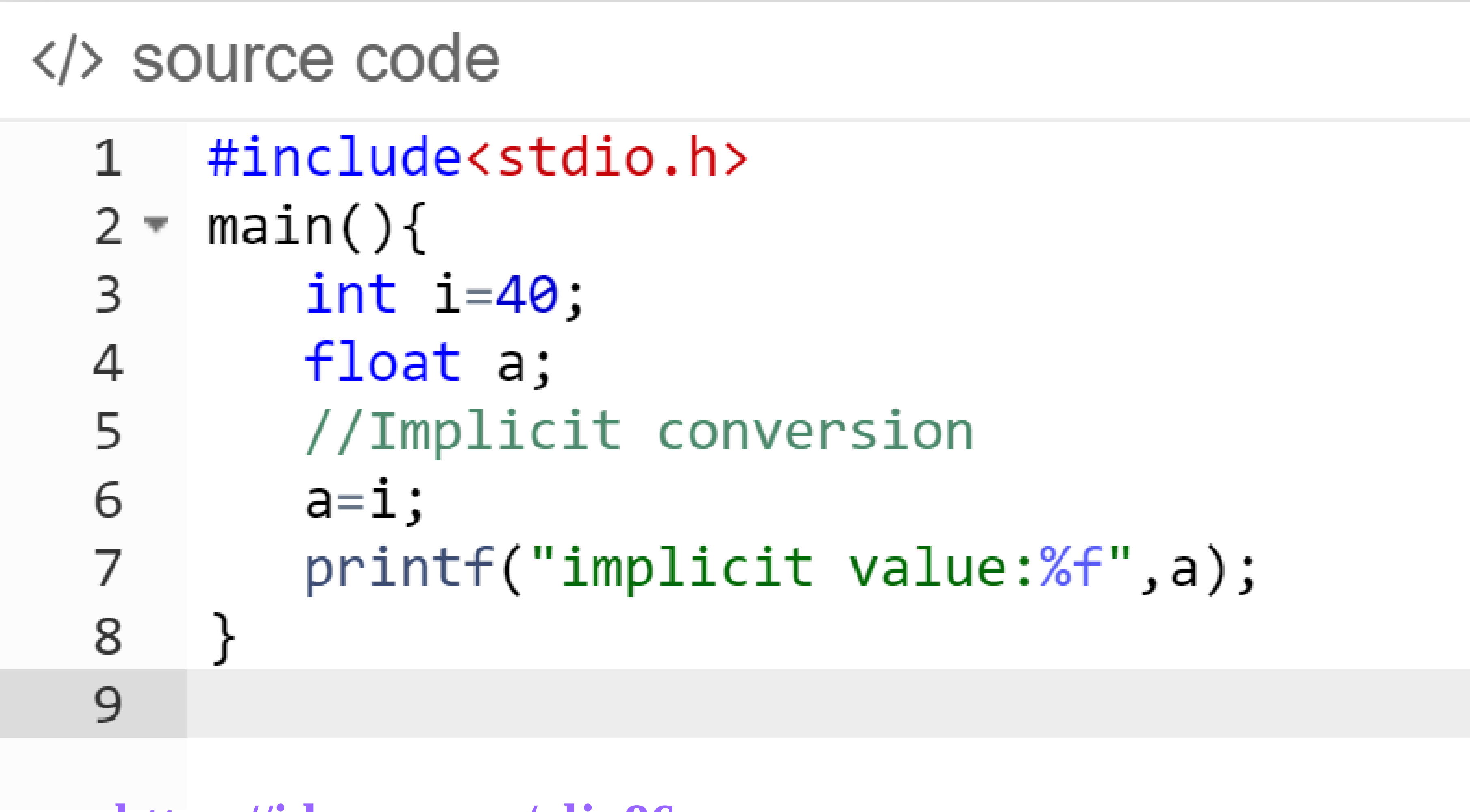
Dr. R. Balasubramanian
Professor
Department of Computer Science and Engineering
Mehta Family School of Data Science and Artificial Intelligence
Indian Institute of Technology Roorkee
Roorkee 247 667

*bala@cs.iitr.ac.in*
*https://faculty.iitr.ac.in/cs/bala/*

# Type casing


Higher Data Type
long double
double
float
Long
int
short
char
No Loss in Data
Loss in Data
Lower Data Type

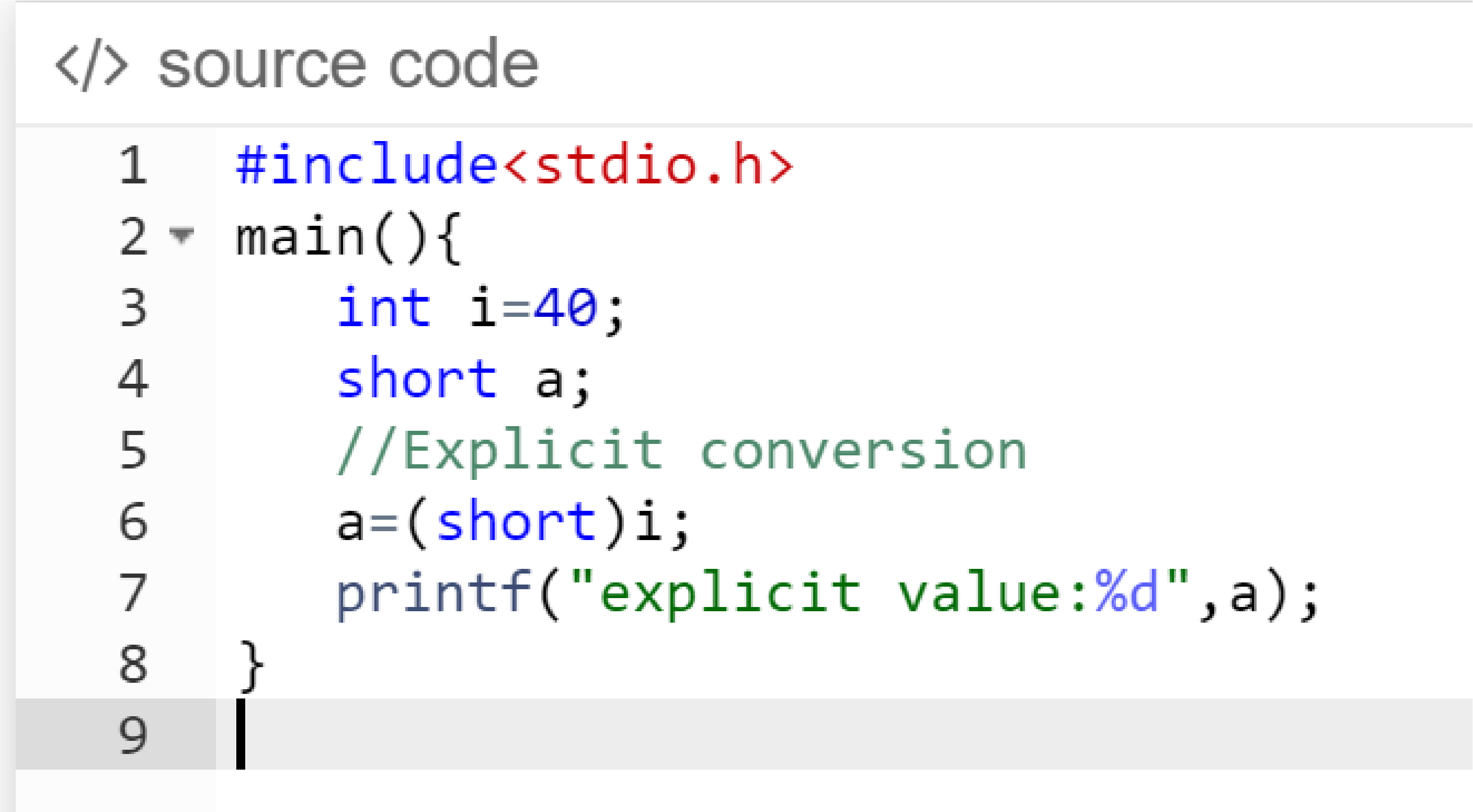
# Implicit casting



```
#include<stdio.h>
main(){
    int i=40;
    float a;
    //Implicit conversion
    a=i;
    printf("implicit value:%f",a);
}

```

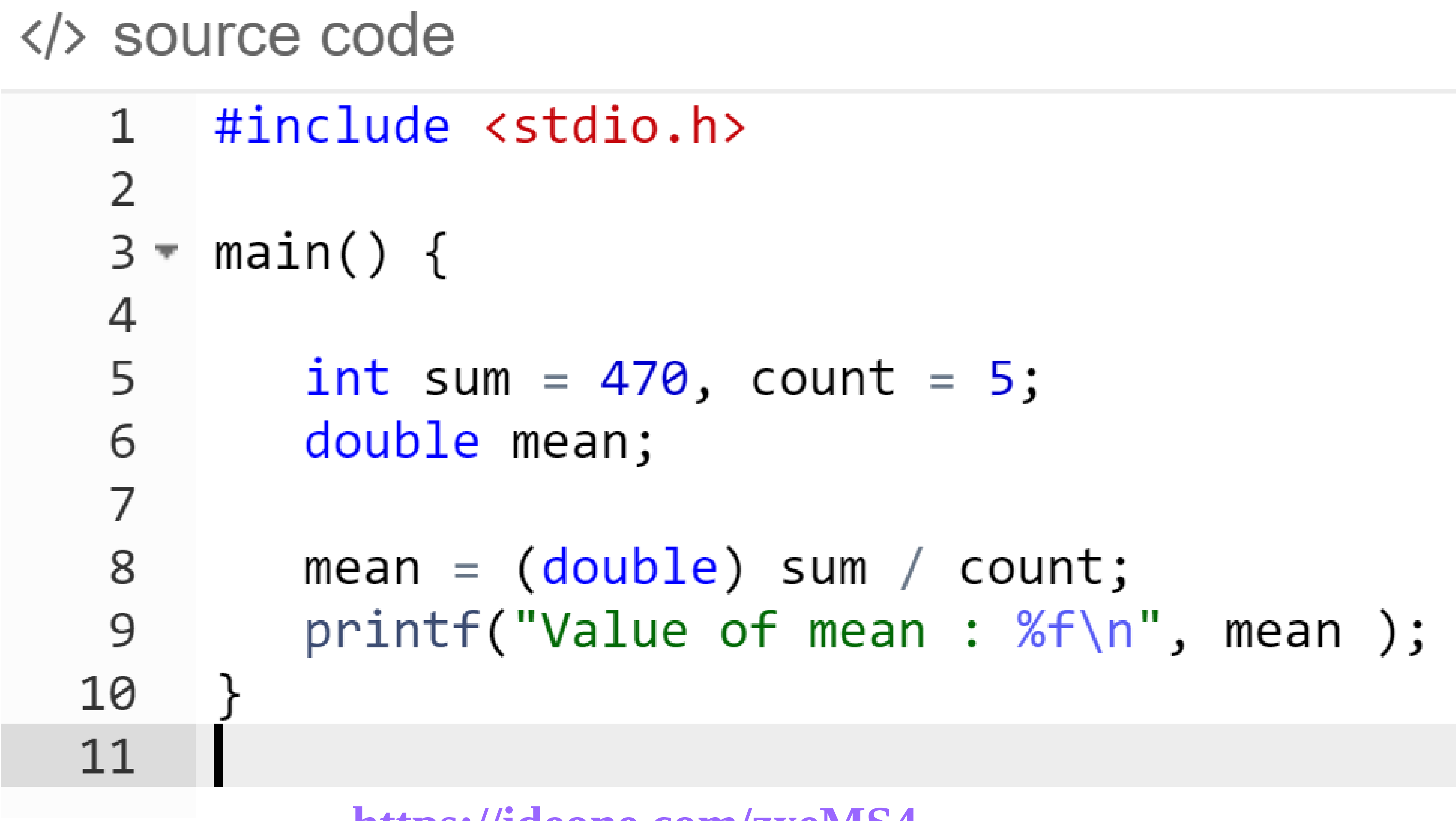
https://ideone.com/vljn86

# Explicit casting



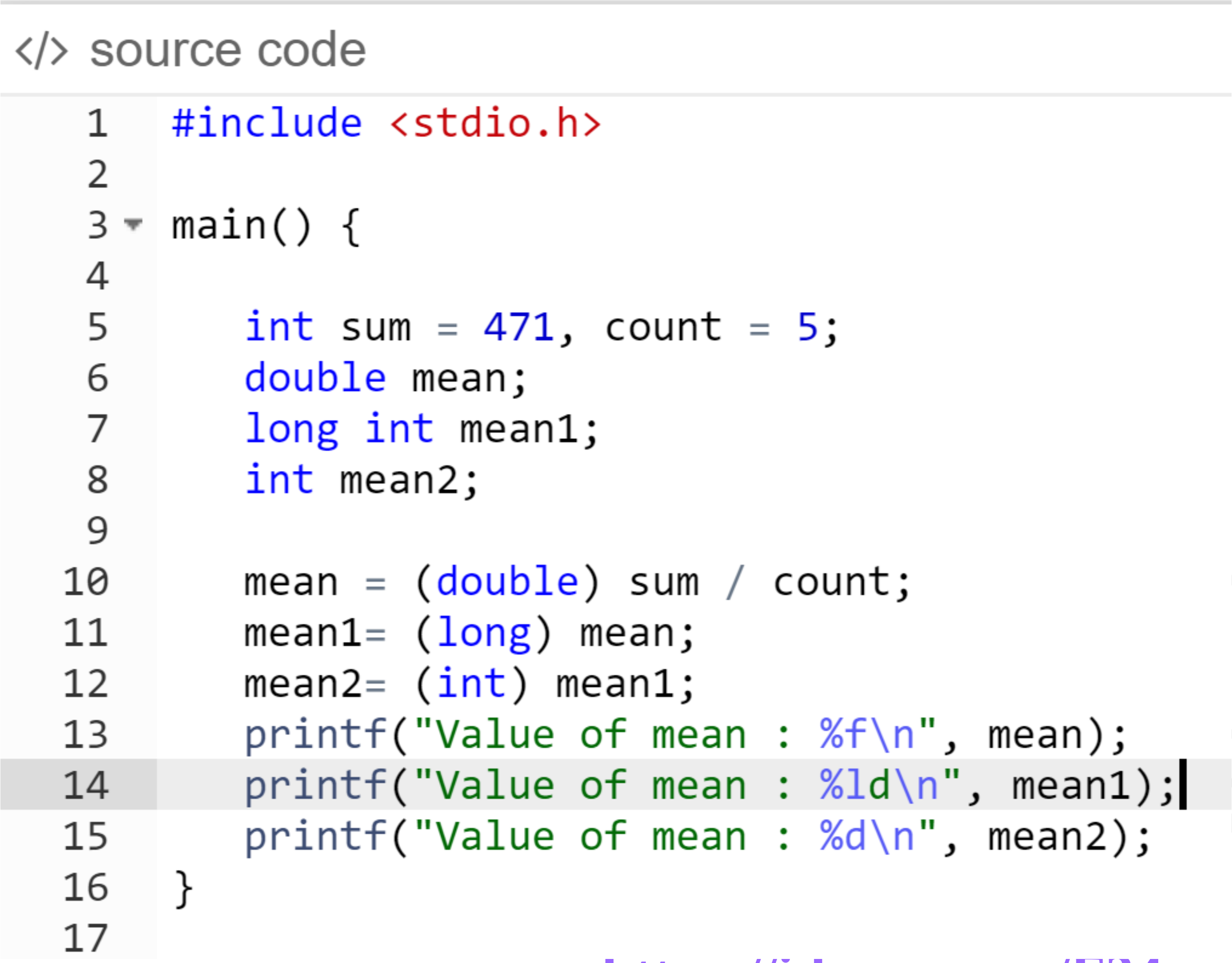
```c
#include<stdio.h>
main(){
    int i=40;
    short a;
    //Explicit conversion
    a=(short)i;
    printf("explicit value:%d",a);
}
```

https://ideone.com/rWWOUl



```c
#include <stdio.h>

main() {

    int sum = 470, count = 5;
    double mean;

    mean = (double) sum / count;
    printf("Value of mean : %f\n", mean );
}

```

https://ideone.com/zxeMS4



</> source code

```c
#include <stdio.h>

main() {

    int sum = 471, count = 5;
    double mean;
    long int mean1;
    int mean2;

    mean = (double) sum / count;
    mean1= (long) mean;
    mean2= (int) mean1;
    printf("Value of mean : %f\n", mean);
    printf("Value of mean : %ld\n", mean1);
    printf("Value of mean : %d\n", mean2);
}

```

https://ideone.com/F34xod

# Operators in C

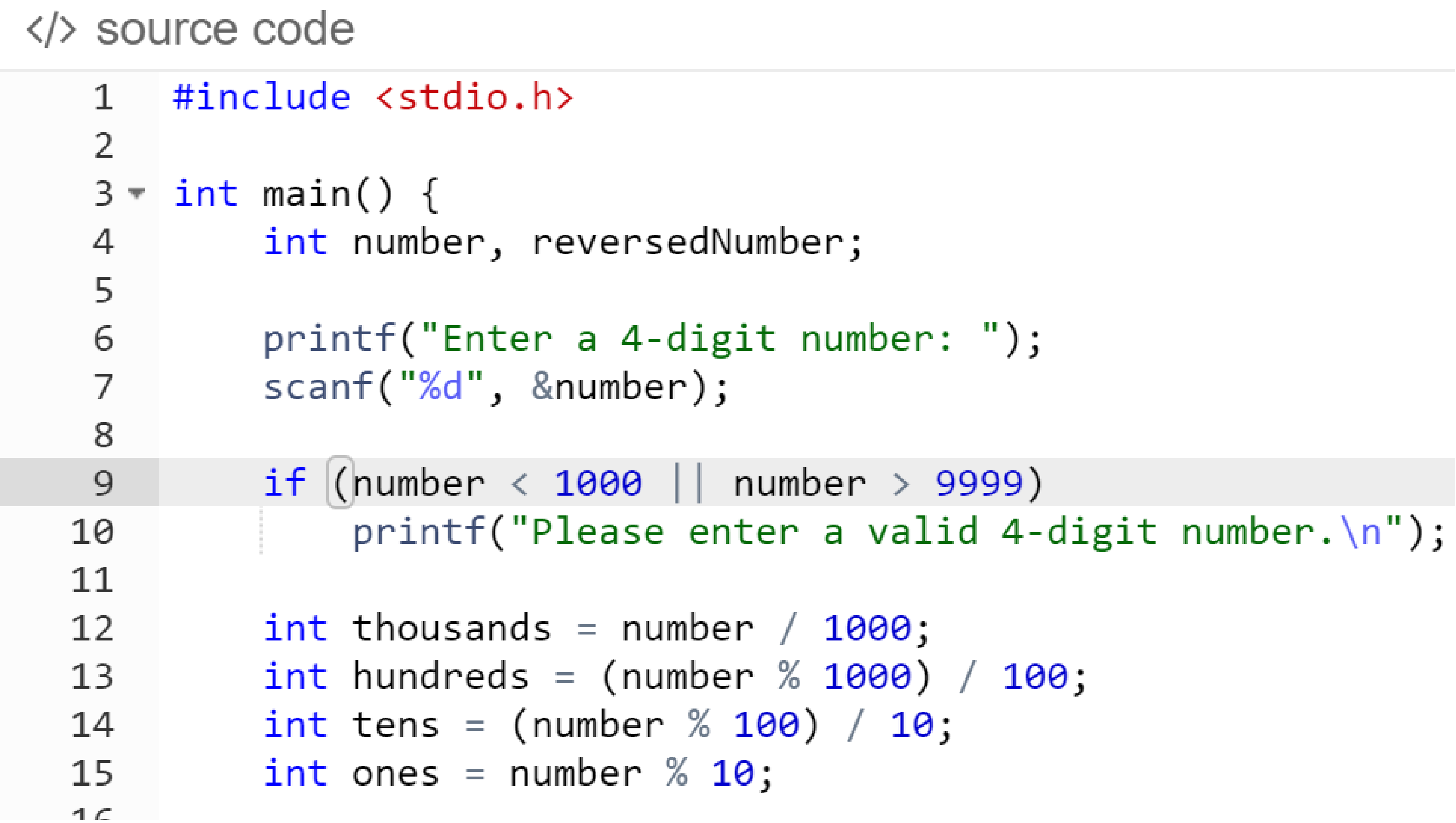
- Arithmetic Operators
- Relational Operators
- Shift Operators
- Logical Operators
- Bitwise Operators
- Ternary or Conditional Operators
- Assignment Operator
- Misc Operator

# Arithmetic Operators

| Operator | Use of Operator |
|---|---|
| + | Use to Add Two Numbers and Also used to Concatenate two strings |
| - | Used for Subtraction |
| * | Used to multiply numbers |
| / | Used for Division |
| % | Used for Finding Mod (Remainder Operator) |

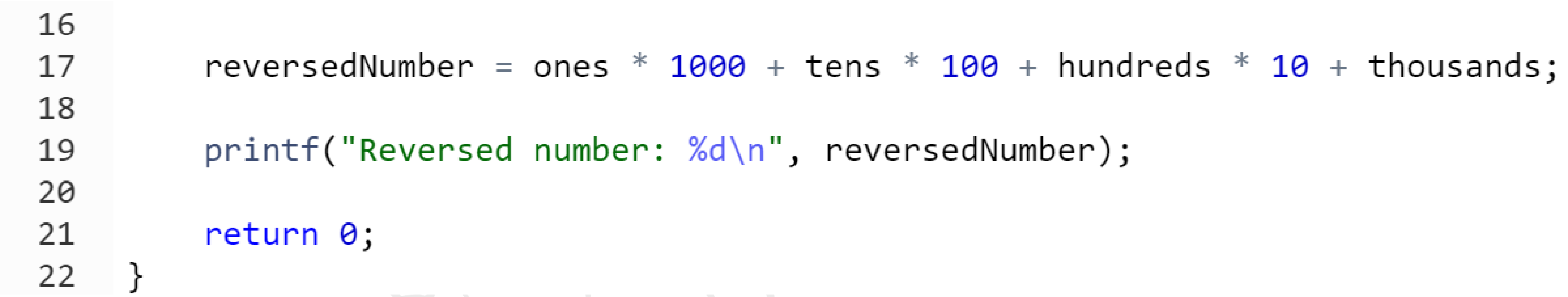
# Arithmetic Operators

- Write a C Program to reverse 4 digit number using arithmetic operator



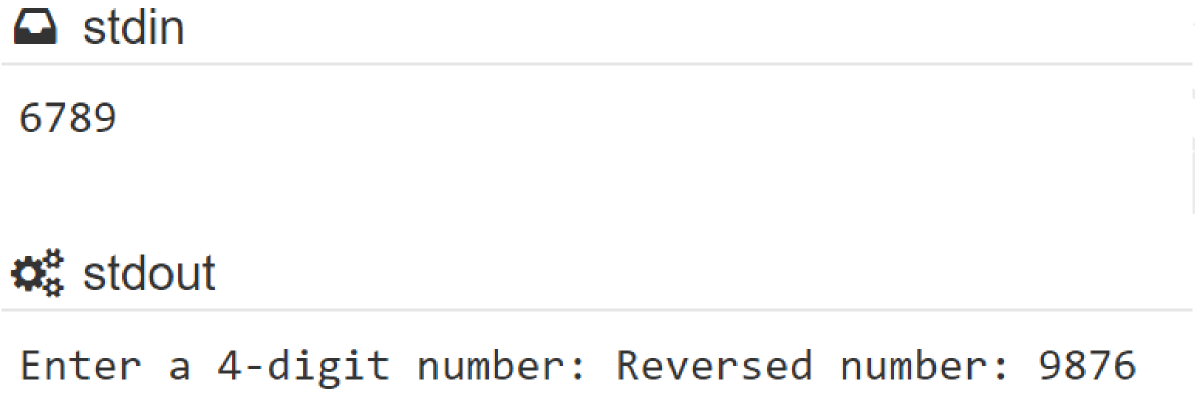
</> source code

```c
#include <stdio.h>

int main() {
    int number, reversedNumber;

    printf("Enter a 4-digit number: ");
    scanf("%d", &number);

    if (number < 1000 || number > 9999)
        printf("Please enter a valid 4-digit number.\n");

    int thousands = number / 1000;
    int hundreds = (number % 1000) / 100;
    int tens = (number % 100) / 10;
    int ones = number % 10;
```



```

    reversedNumber = ones * 1000 + tens * 100 + hundreds * 10 + thousands;

    printf("Reversed number: %d\n", reversedNumber);

    return 0;
}
```

[https://ideone.com/N2F860](https://ideone.com/N2F860)



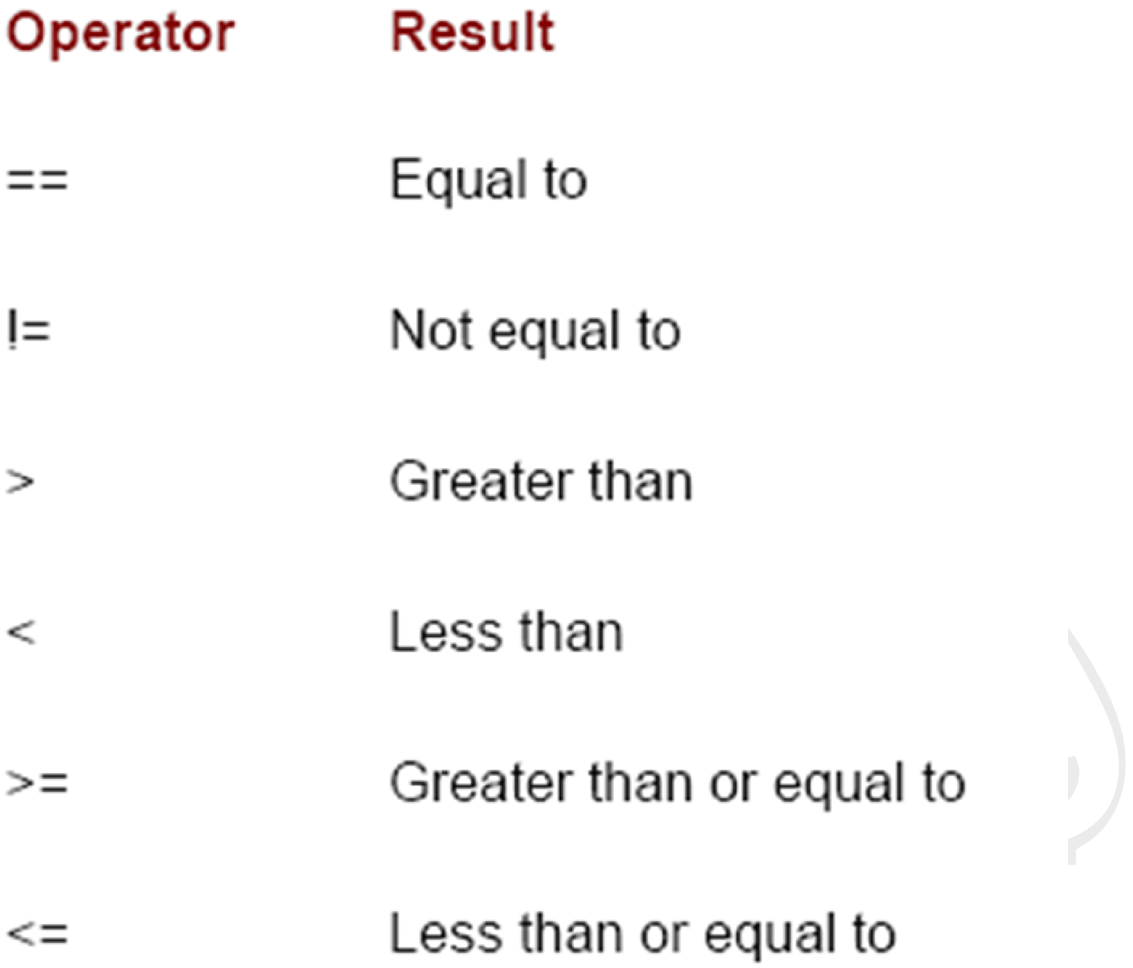
stdin

```
6789
```

stdout

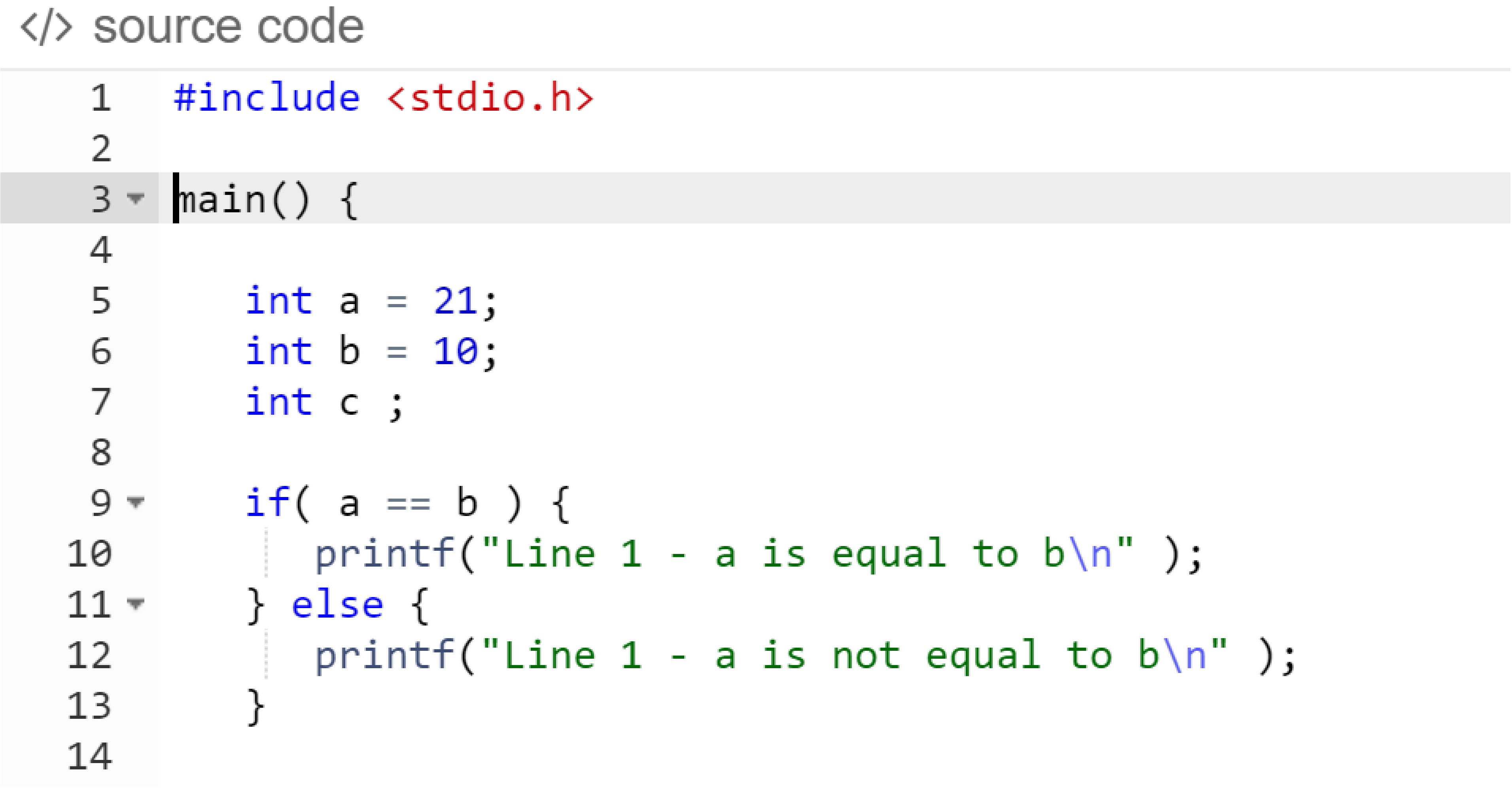
```
Enter a 4-digit number: Reversed number: 9876
```

# Relational Operator



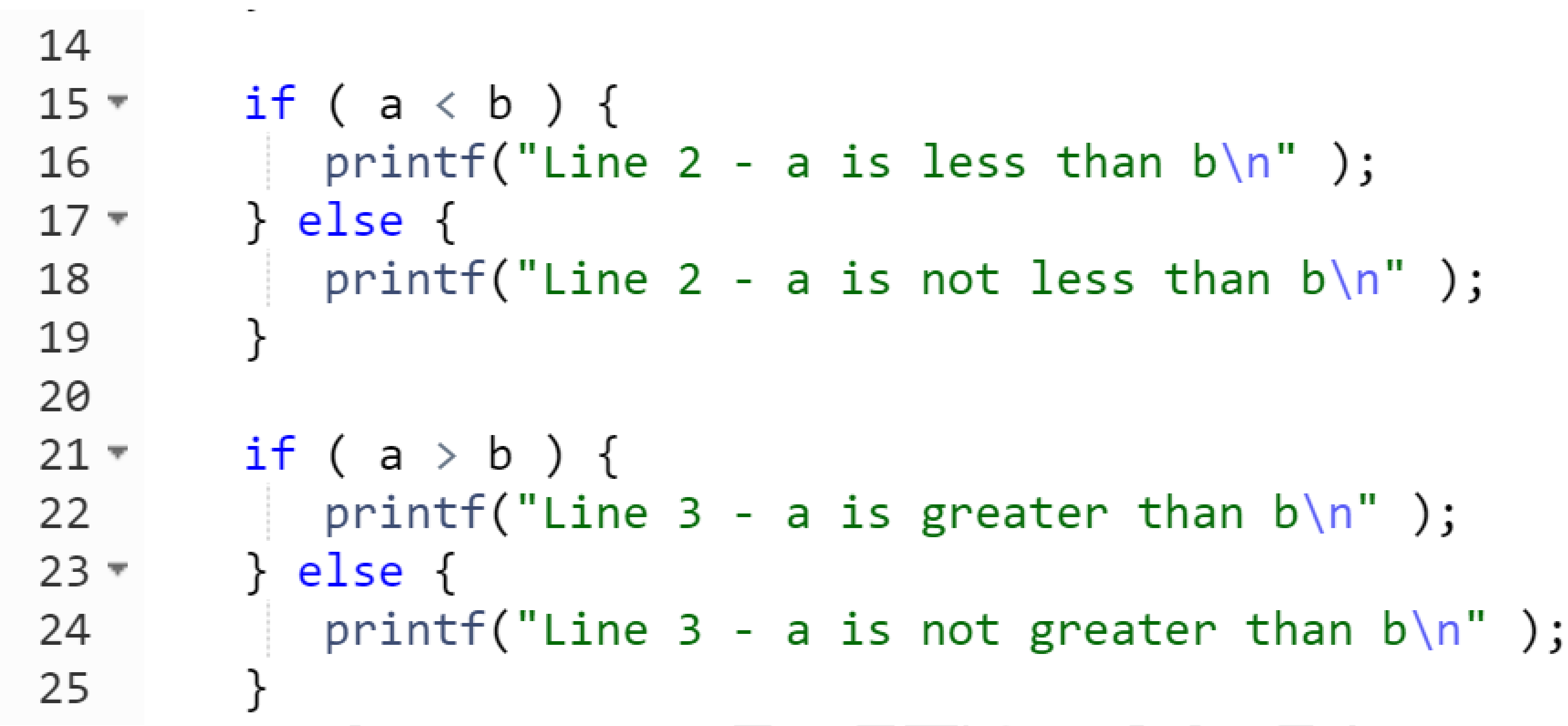
| Operator | Result |
|---|---|
| == | Equal to |
| != | Not equal to |
| > | Greater than |
| < | Less than |
| >= | Greater than or equal to |
| <= | Less than or equal to |

# Relational Operator



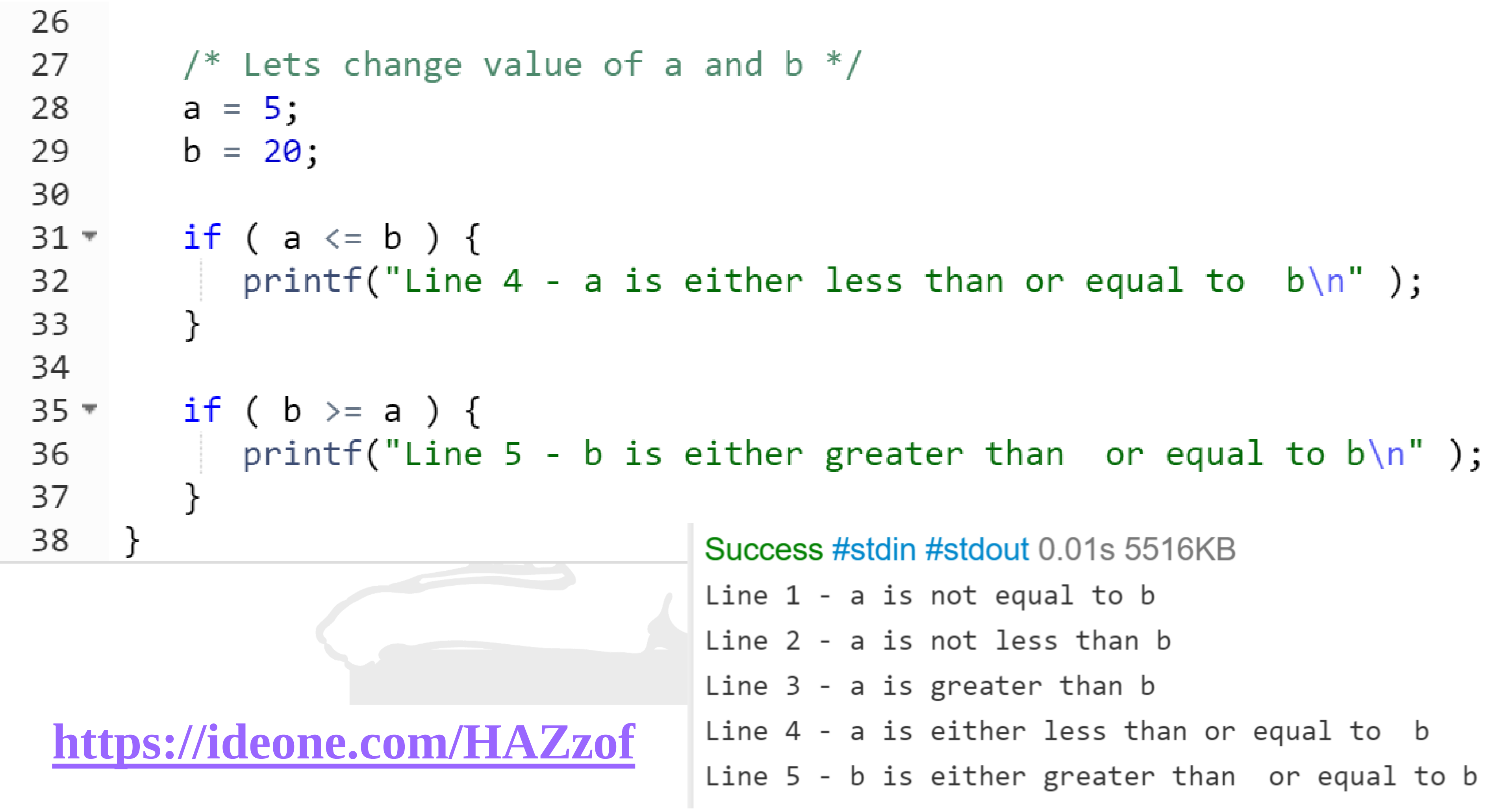
```c
#include <stdio.h>

main() {

    int a = 21;
    int b = 10;
    int c ;

    if( a == b ) {
        printf("Line 1 - a is equal to b\n" );
    } else {
        printf("Line 1 - a is not equal to b\n" );
    }

```

# Relational Operator



```
if ( a < b ) {
    printf("Line 2 - a is less than b\n" );
} else {
    printf("Line 2 - a is not less than b\n" );
}

if ( a > b ) {
    printf("Line 3 - a is greater than b\n" );
} else {
    printf("Line 3 - a is not greater than b\n" );
}
```

# Relational Operator



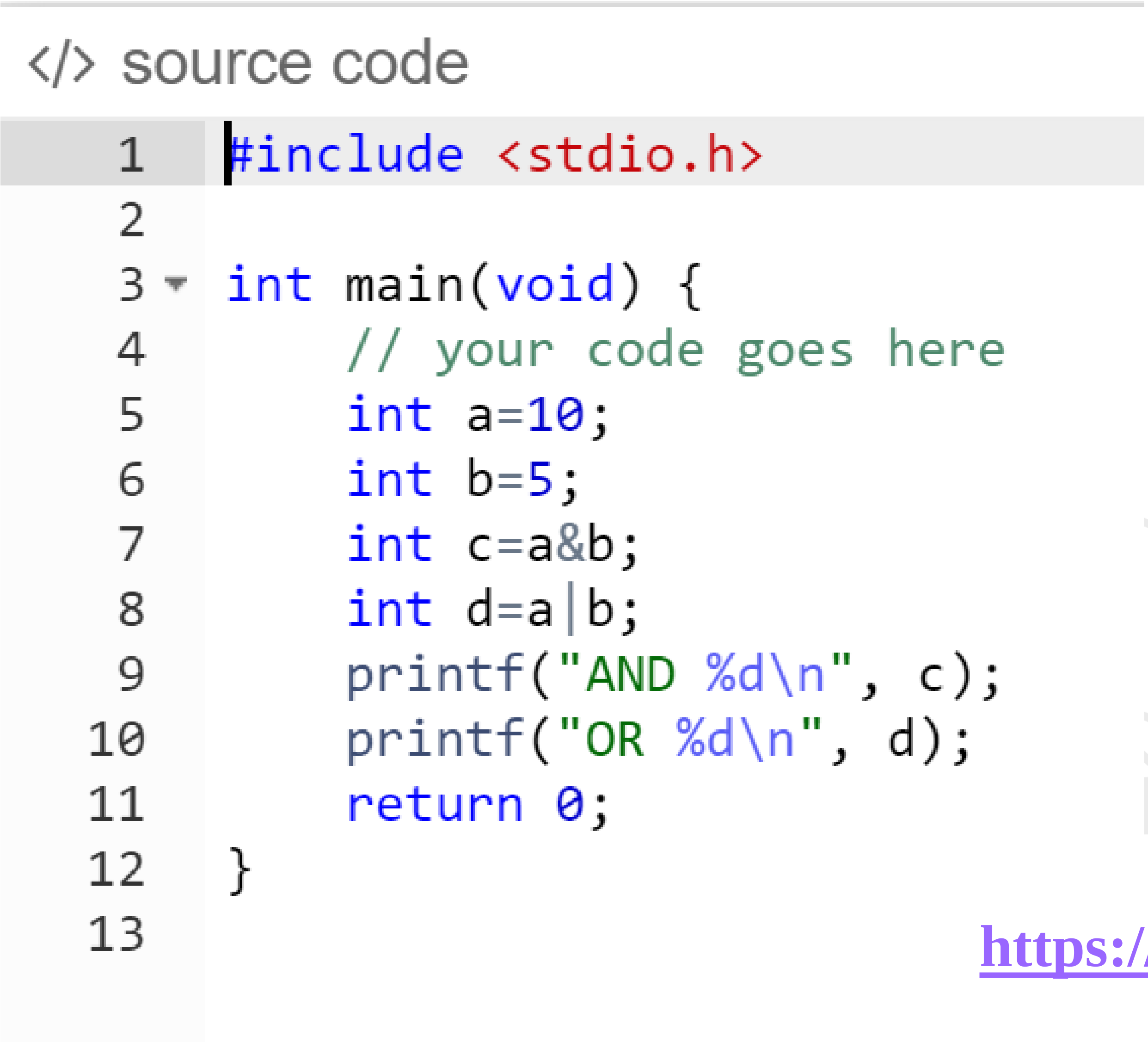
```c

    /* Lets change value of a and b */
    a = 5;
    b = 20;

    if ( a <= b ) {
       printf("Line 4 - a is either less than or equal to  b\n" );
    }

    if ( b >= a ) {
       printf("Line 5 - b is either greater than  or equal to b\n" );
    }
}
```

Success #stdin #stdout 0.01s 5516KB

```
Line 1 - a is not equal to b
Line 2 - a is not less than b
Line 3 - a is greater than b
Line 4 - a is either less than or equal to  b
Line 5 - b is either greater than  or equal to b
```

https://ideone.com/HAZzof

# Bitwise operators



</> source code

```c
#include <stdio.h>

int main(void) {
    // your code goes here
    int a=10;
    int b=5;
    int c=a&b;
    int d=a|b;
    printf("AND %d\n", c);
    printf("OR %d\n", d);
    return 0;
}

```

https://ideone.com/r7SMJF

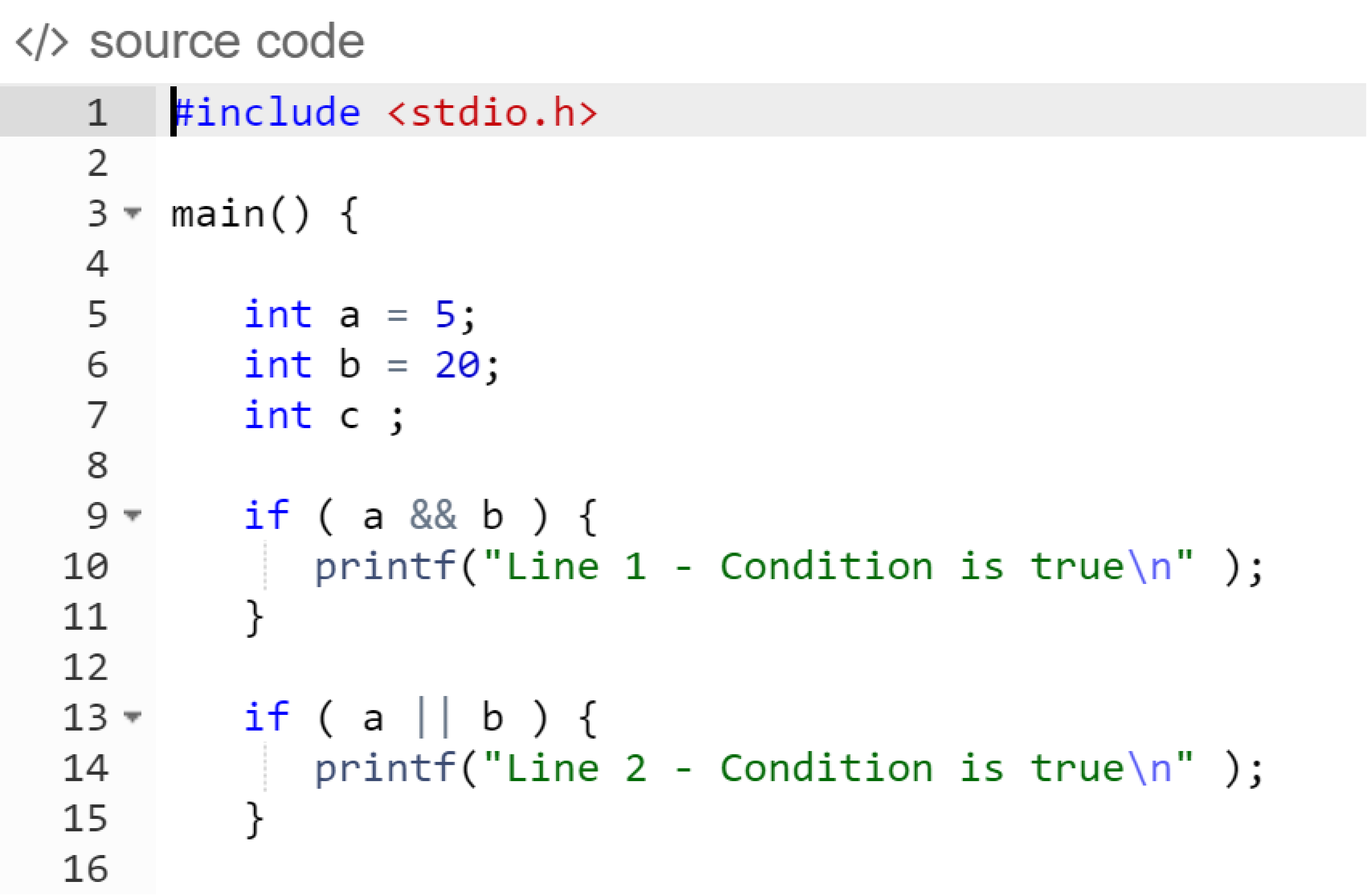
# Powers of 2

- Write a C program to find the maximum of three numbers

# Logical (Conditional) Operators

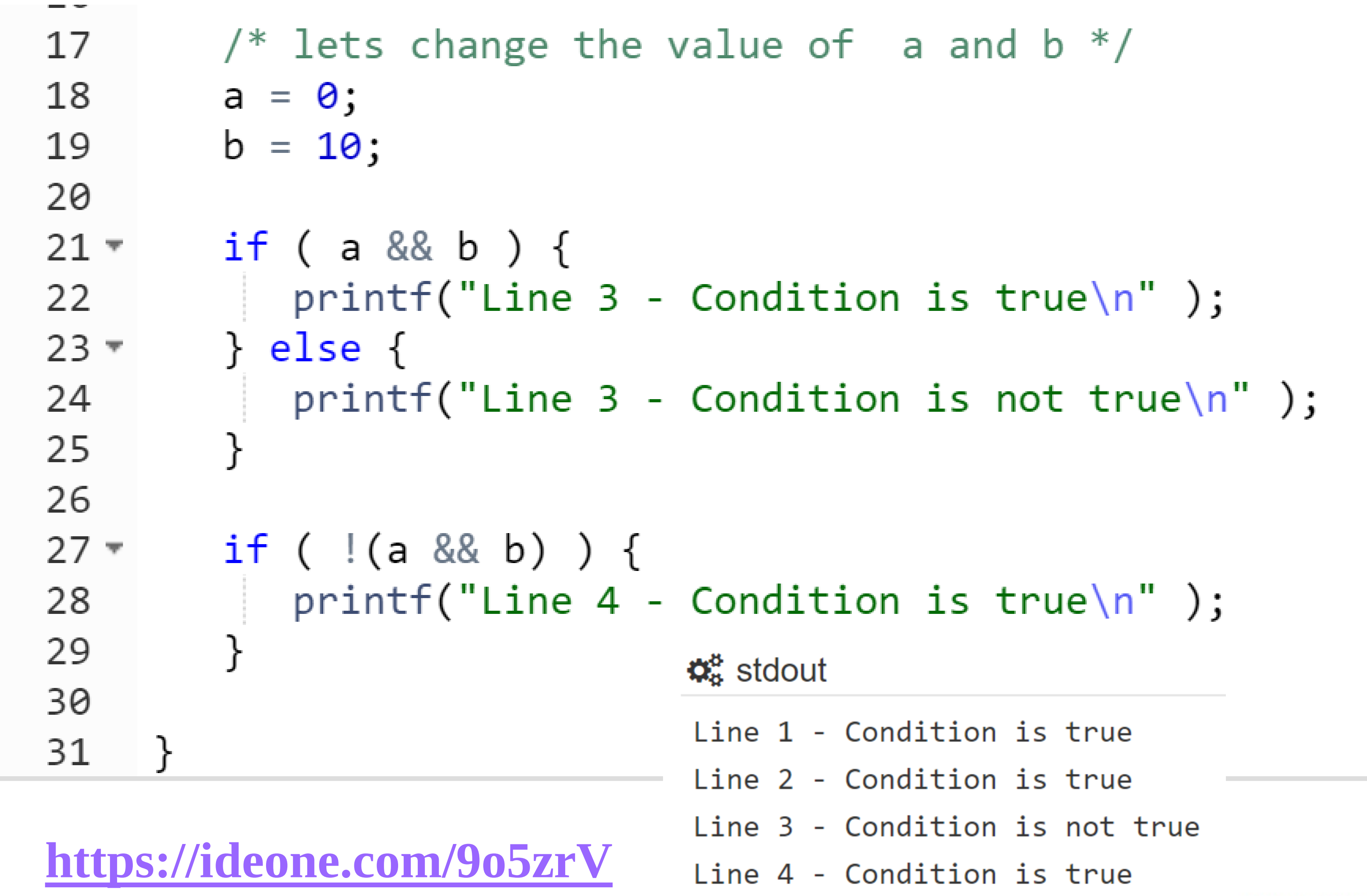
| Operator | Use of Operator |
|---|---|
| && | Logical-AND |
| \|\| | Logical-OR |
| ! | Logical- NOT |

# Logical (Conditional) Operators



</> source code

```c
#include <stdio.h>

main() {

    int a = 5;
    int b = 20;
    int c ;

    if ( a && b ) {
        printf("Line 1 - Condition is true\n" );
    }

    if ( a || b ) {
        printf("Line 2 - Condition is true\n" );
    }

```



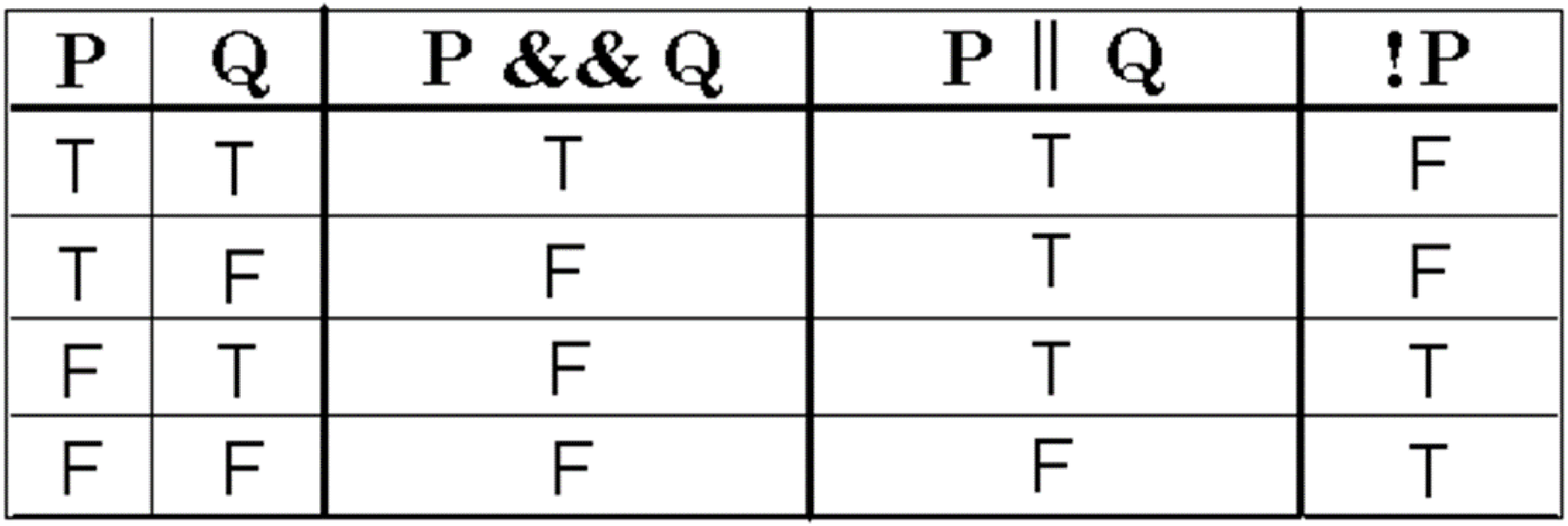
```c
    /* lets change the value of  a and b */
    a = 0;
    b = 10;

    if ( a && b ) {
        printf("Line 3 - Condition is true\n" );
    } else {
        printf("Line 3 - Condition is not true\n" );
    }

    if ( !(a && b) ) {
        printf("Line 4 - Condition is true\n" );
    }

}
```

stdout

```
Line 1 - Condition is true
Line 2 - Condition is true
Line 3 - Condition is not true
Line 4 - Condition is true
```

https://ideone.com/9o5zrV

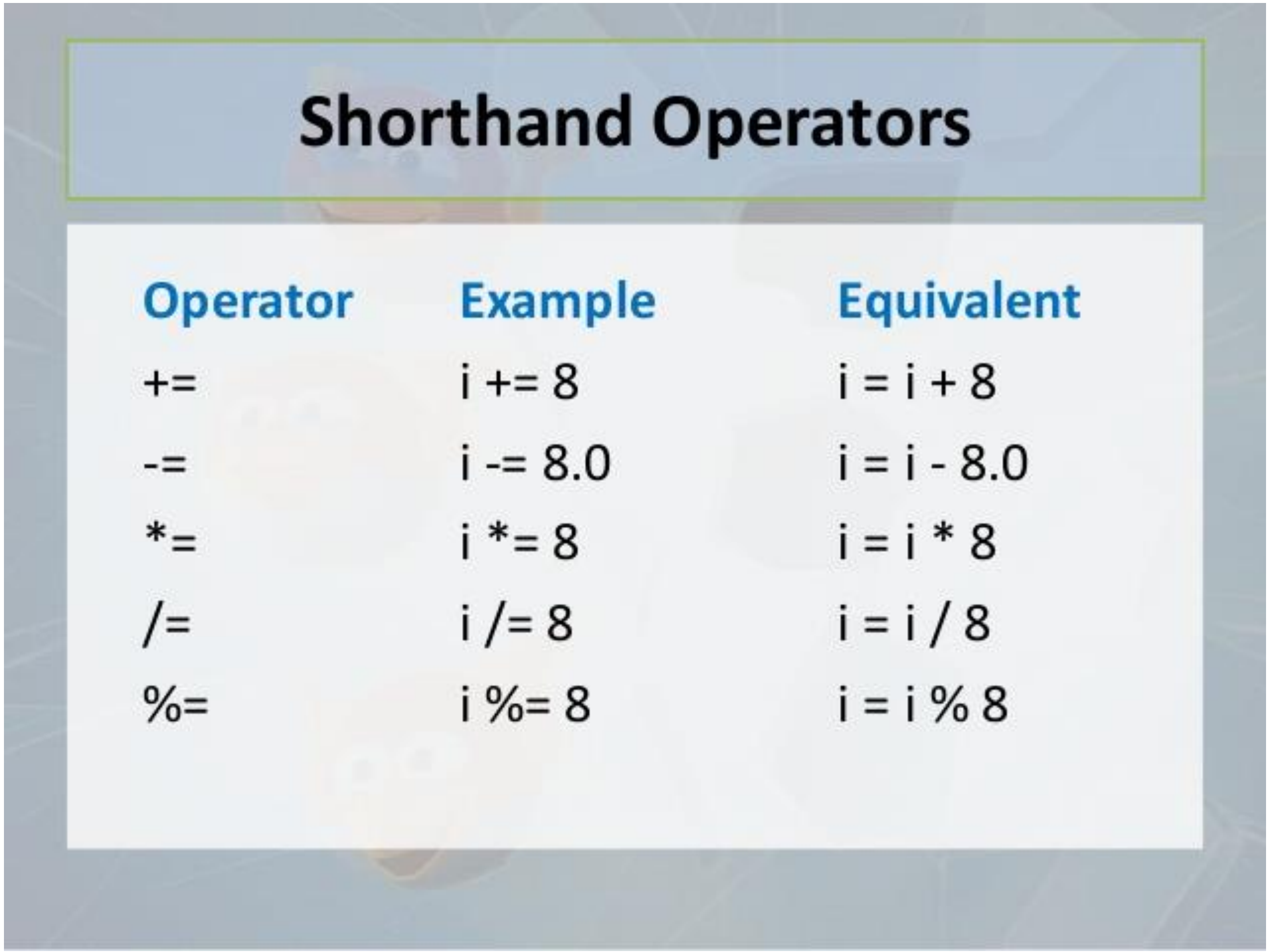
# Truth Table



| P | Q | P && Q | P \|\| Q | !P |
|---|---|---|---|---|
| T | T | T | T | F |
| T | F | F | T | F |
| F | T | F | T | T |
| F | F | F | F | T |

# Assignment Operators

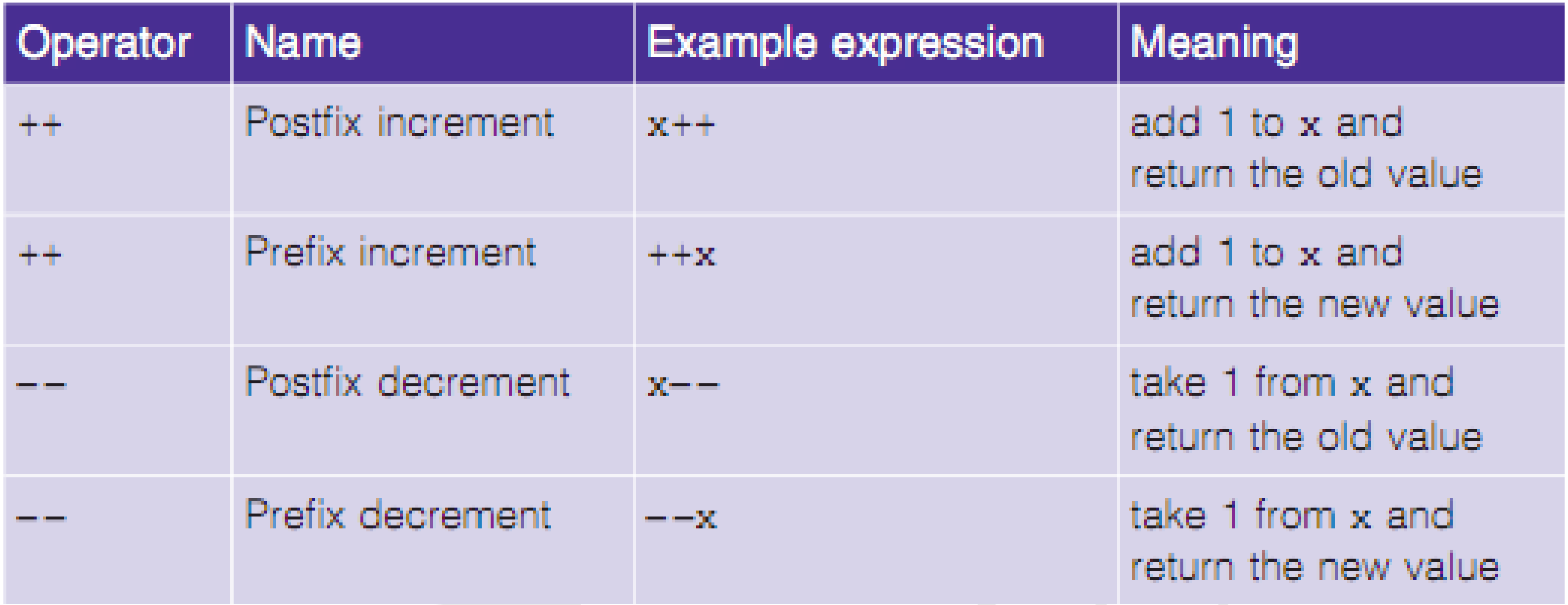
```
Sum=a+b;
```



## Shorthand Operators

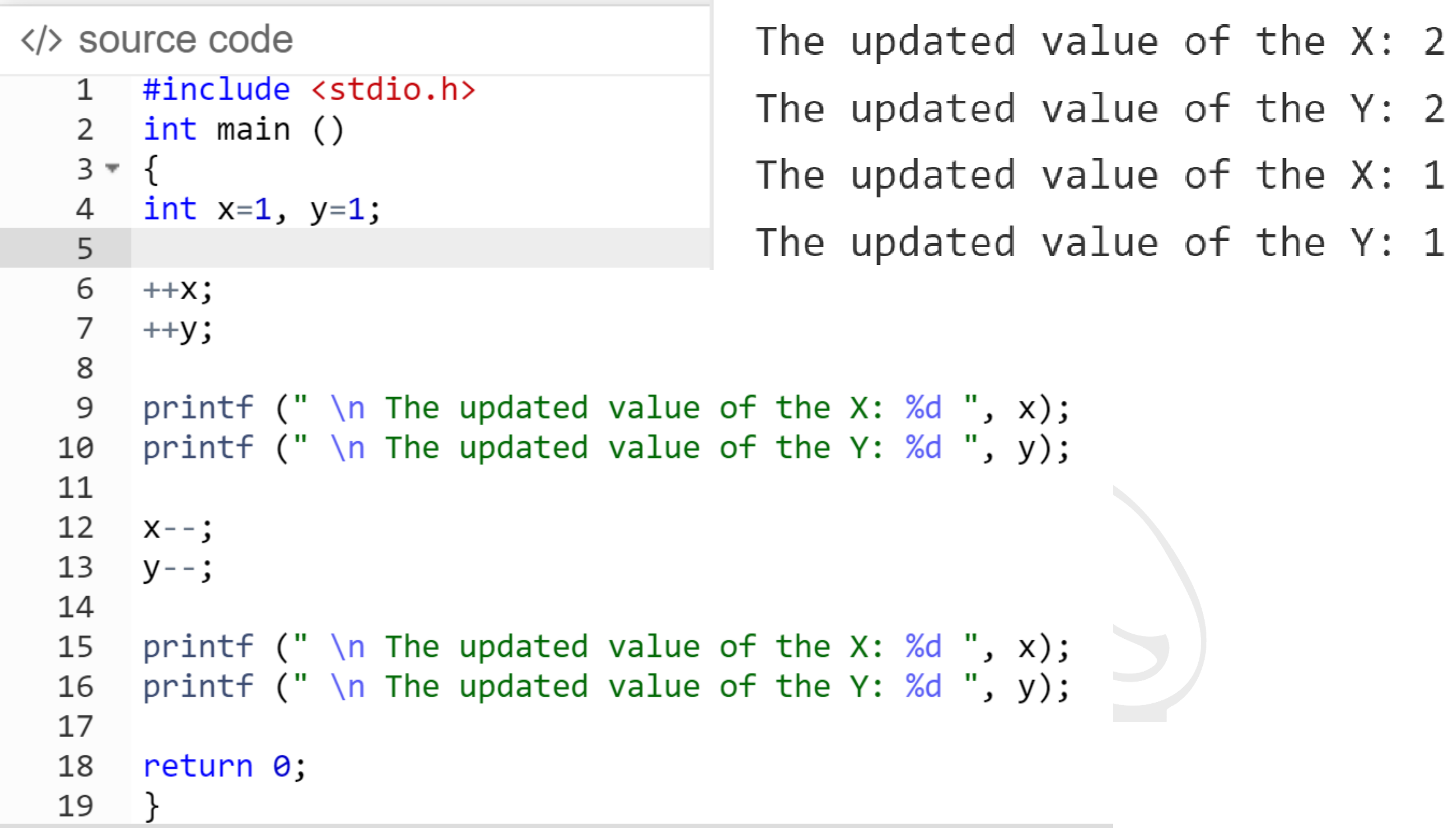
| Operator | Example | Equivalent |
|---|---|---|
| += | i += 8 | i = i + 8 |
| -= | i -= 8.0 | i = i - 8.0 |
| *= | i *= 8 | i = i * 8 |
| /= | i /= 8 | i = i / 8 |
| %= | i %= 8 | i = i % 8 |

# Increment and Decrement Operators



| Operator | Name | Example expression | Meaning |
|---|---|---|---|
| ++ | Postfix increment | x++ | add 1 to x and return the old value |
| ++ | Prefix increment | ++x | add 1 to x and return the new value |
| -- | Postfix decrement | x-- | take 1 from x and return the old value |
| -- | Prefix decrement | --x | take 1 from x and return the new value |

# Increment and Decrement Operators

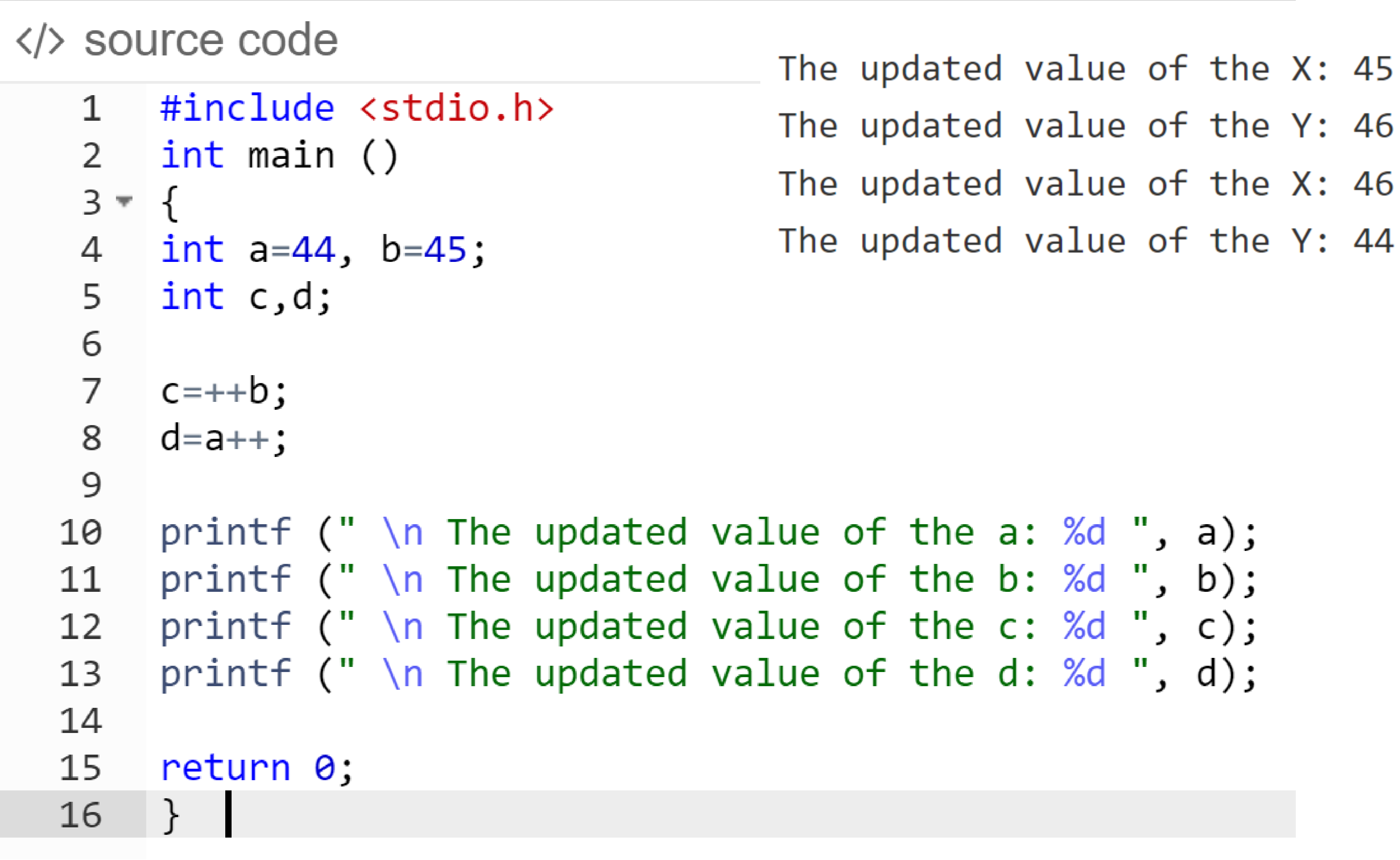


```c
#include <stdio.h>
int main ()
{
int x=1, y=1;

++x;
++y;

printf (" \n The updated value of the X: %d ", x);
printf (" \n The updated value of the Y: %d ", y);

x--;
y--;

printf (" \n The updated value of the X: %d ", x);
printf (" \n The updated value of the Y: %d ", y);

return 0;
}
```

```
The updated value of the X: 2
The updated value of the Y: 2
The updated value of the X: 1
The updated value of the Y: 1
```

https://ideone.com/N3eJ6d

# Increment Operator

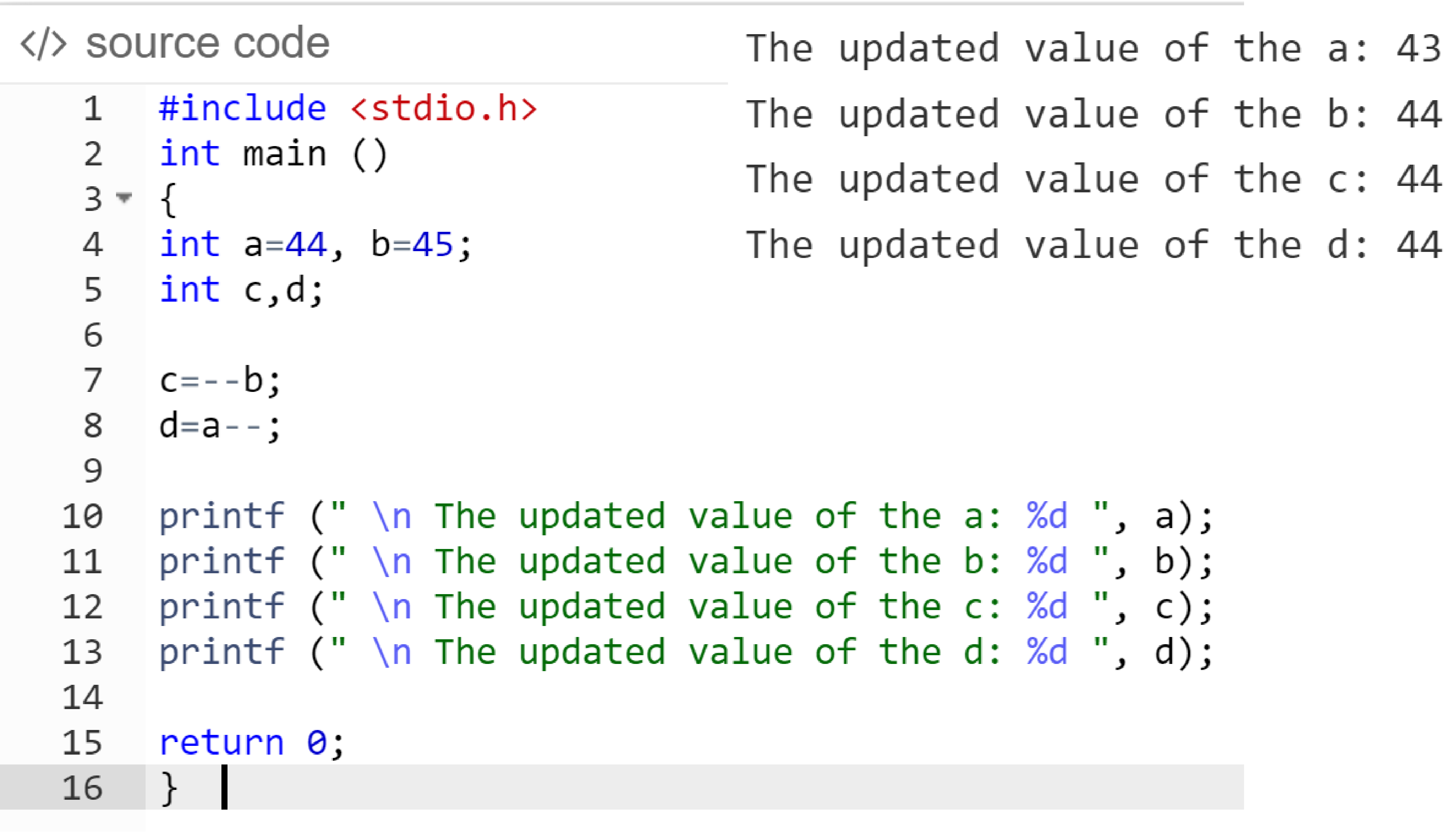


```c
#include <stdio.h>
int main ()
{
int a=44, b=45;
int c,d;

c=++b;
d=a++;

printf (" \n The updated value of the a: %d ", a);
printf (" \n The updated value of the b: %d ", b);
printf (" \n The updated value of the c: %d ", c);
printf (" \n The updated value of the d: %d ", d);

return 0;
}
```

```
The updated value of the X: 45
The updated value of the Y: 46
The updated value of the X: 46
The updated value of the Y: 44
```

https://ideone.com/gdJH16

# Decrement Operator



```c
#include <stdio.h>
int main ()
{
int a=44, b=45;
int c,d;

c=--b;
d=a--;

printf (" \n The updated value of the a: %d ", a);
printf (" \n The updated value of the b: %d ", b);
printf (" \n The updated value of the c: %d ", c);
printf (" \n The updated value of the d: %d ", d);

return 0;
}
```

```
The updated value of the a: 43
The updated value of the b: 44
The updated value of the c: 44
The updated value of the d: 44
```

https://ideone.com/2F80IF

# Thank You!