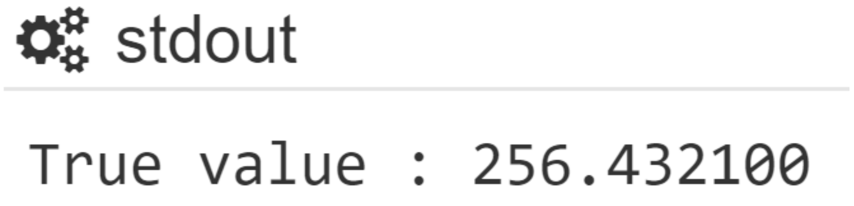
Programming with C and C++

CSC-101 (Lecture 09)

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

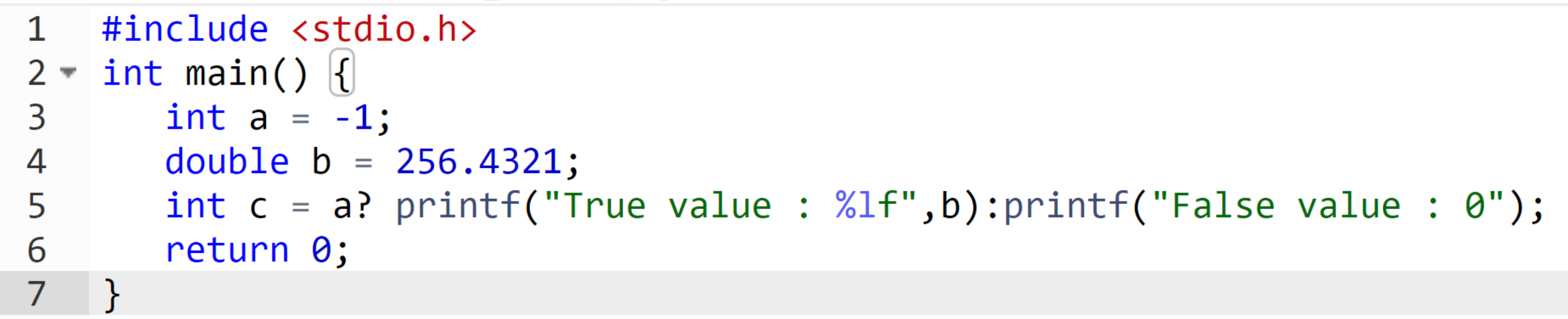
bala@cs.iitr.ac.in
https://faculty.iitr.ac.in/cs/bala/

Ternary Operator



stdout

```
True value : 256.432100
```



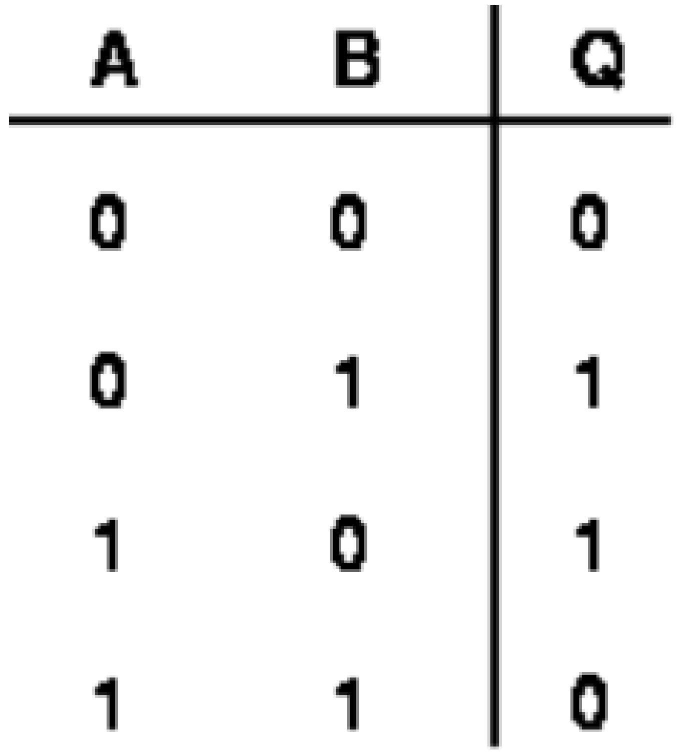
```c
#include <stdio.h>
int main() {
   int a = -1;
   double b = 256.4321;
   int c = a? printf("True value : %lf",b):printf("False value : 0");
   return 0;
}
```

https://ideone.com/3zQ660

Bitwise Operator

Operator	Meaning of operator
&	Bitwise AND operator
|	Bitwise OR operator
^	Bitwise exclusive OR operator
~	One's complement operator (unary operator)
<<	Left shift operator
>>	Right shift operator

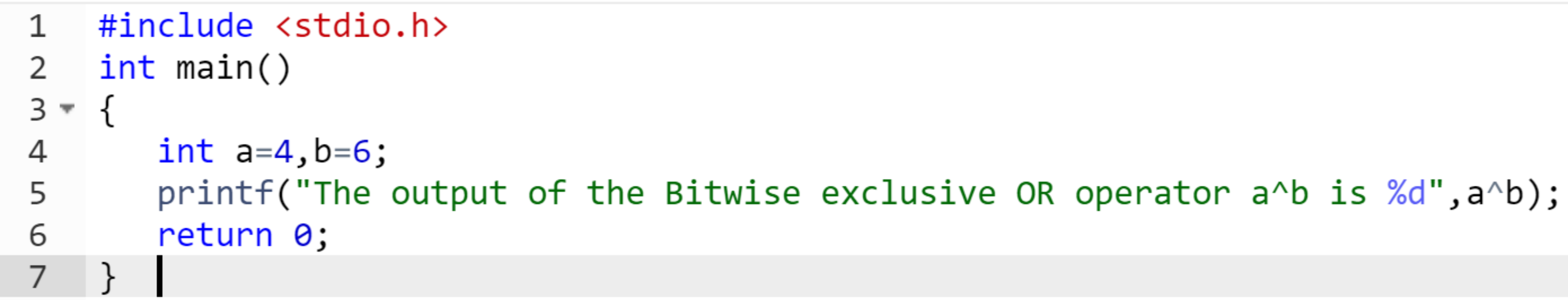
XOR



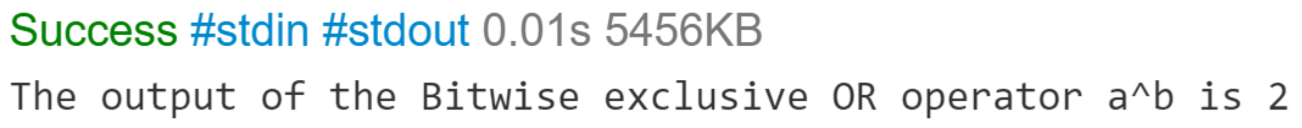
A	B	Q
0	0	0
0	1	1
1	0	1
1	1	0

XOR

Bitwise XOR Operator



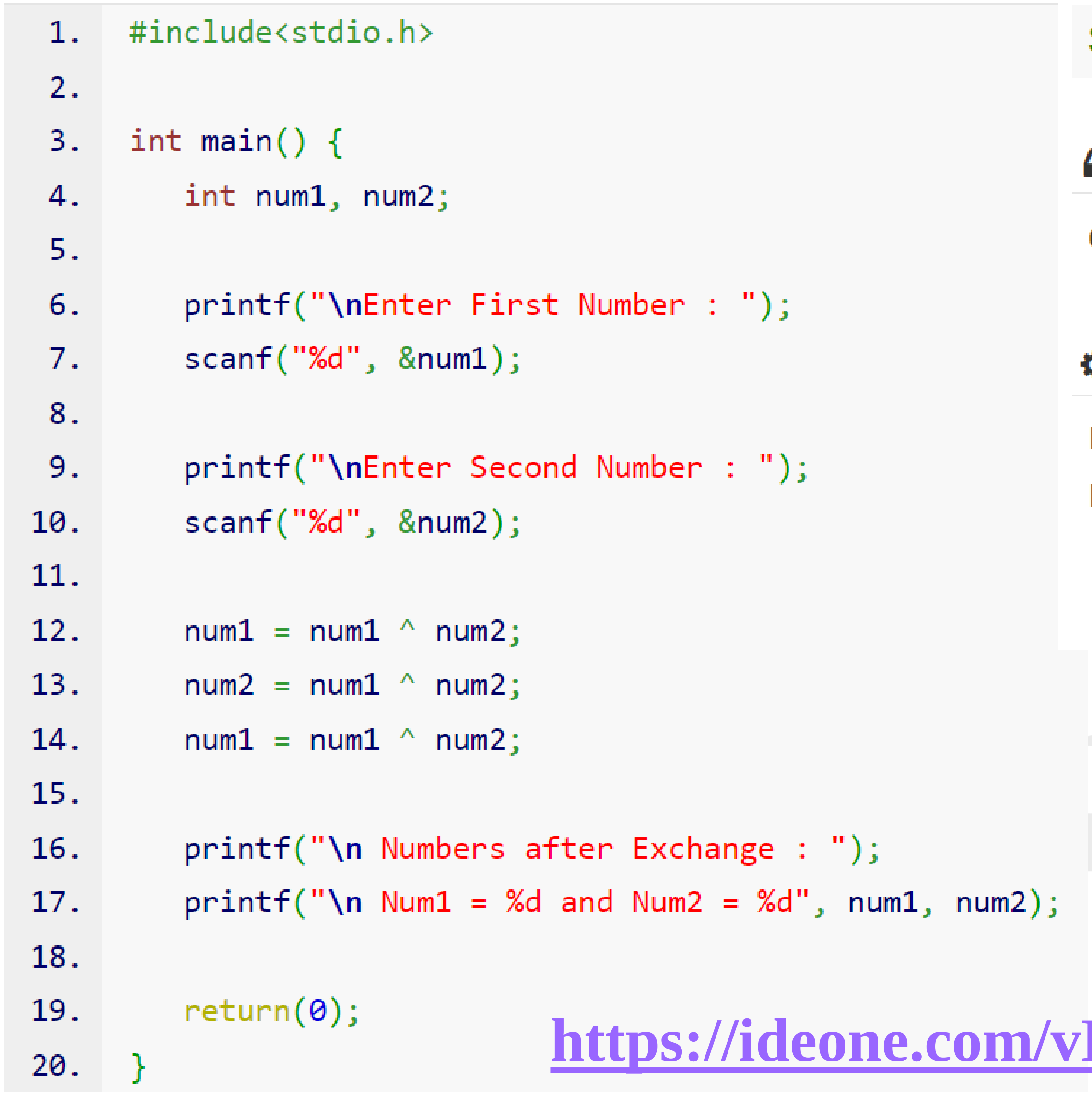
```c
#include <stdio.h>
int main()
{
    int a=4,b=6;
    printf("The output of the Bitwise exclusive OR operator a^b is %d",a^b);
    return 0;
}
```



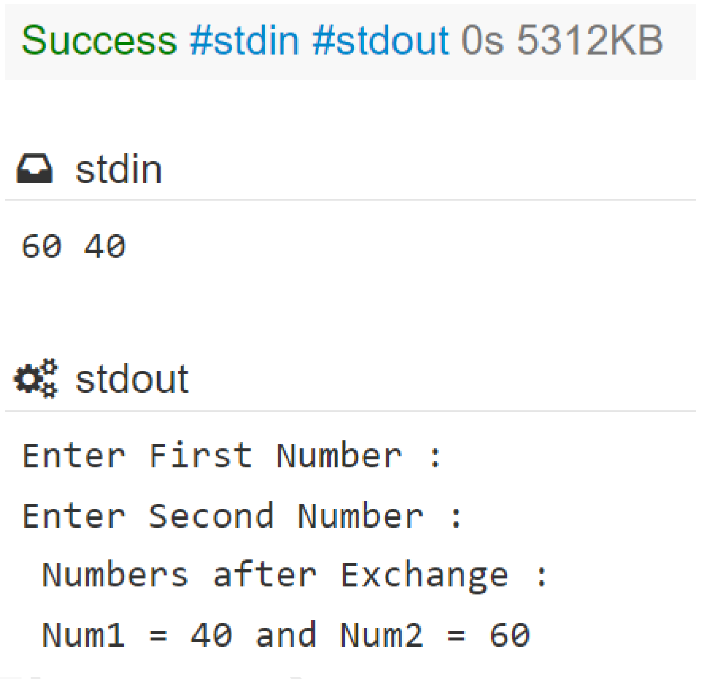
```
Success #stdin #stdout 0.01s 5456KB
The output of the Bitwise exclusive OR operator a^b is 2
```

https://ideone.com/qBNkNI

Swapping two numbers



```c
#include<stdio.h>

int main() {
   int num1, num2;

   printf("\nEnter First Number : ");
   scanf("%d", &num1);

   printf("\nEnter Second Number : ");
   scanf("%d", &num2);

   num1 = num1 ^ num2;
   num2 = num1 ^ num2;
   num1 = num1 ^ num2;

   printf("\n Numbers after Exchange : ");
   printf("\n Num1 = %d and Num2 = %d", num1, num2);

   return(0);
}
```



Success #stdin #stdout 0s 5312KB

stdin

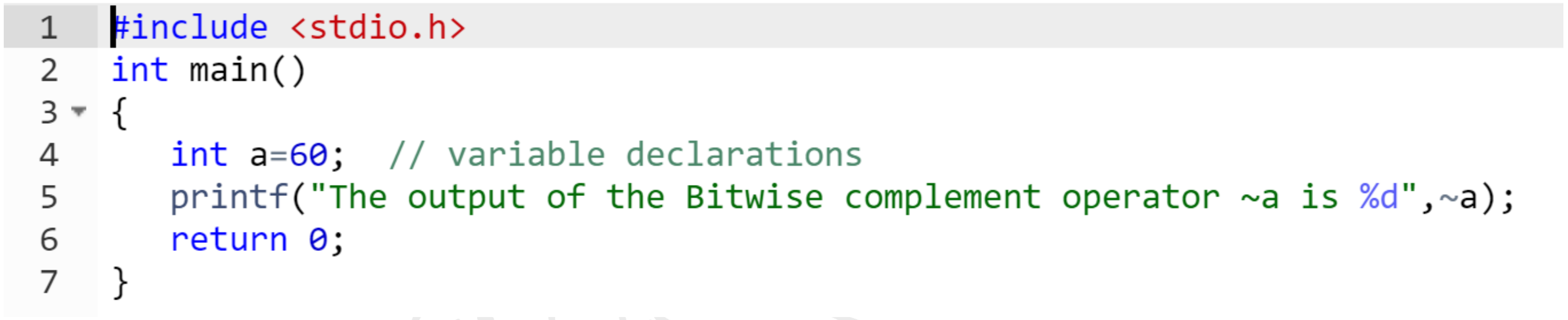
```
60 40
```

stdout

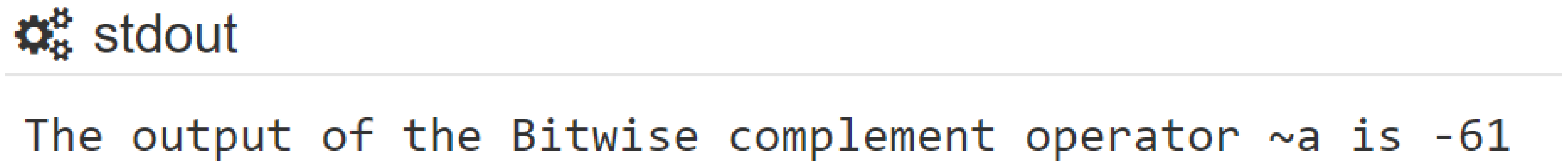
```
Enter First Number : 
Enter Second Number : 
 Numbers after Exchange : 
 Num1 = 40 and Num2 = 60
```

https://ideone.com/vBi6MN

~ Operator



```c
#include <stdio.h>
int main()
{
    int a=60;  // variable declarations
    printf("The output of the Bitwise complement operator ~a is %d",~a);
    return 0;
}
```

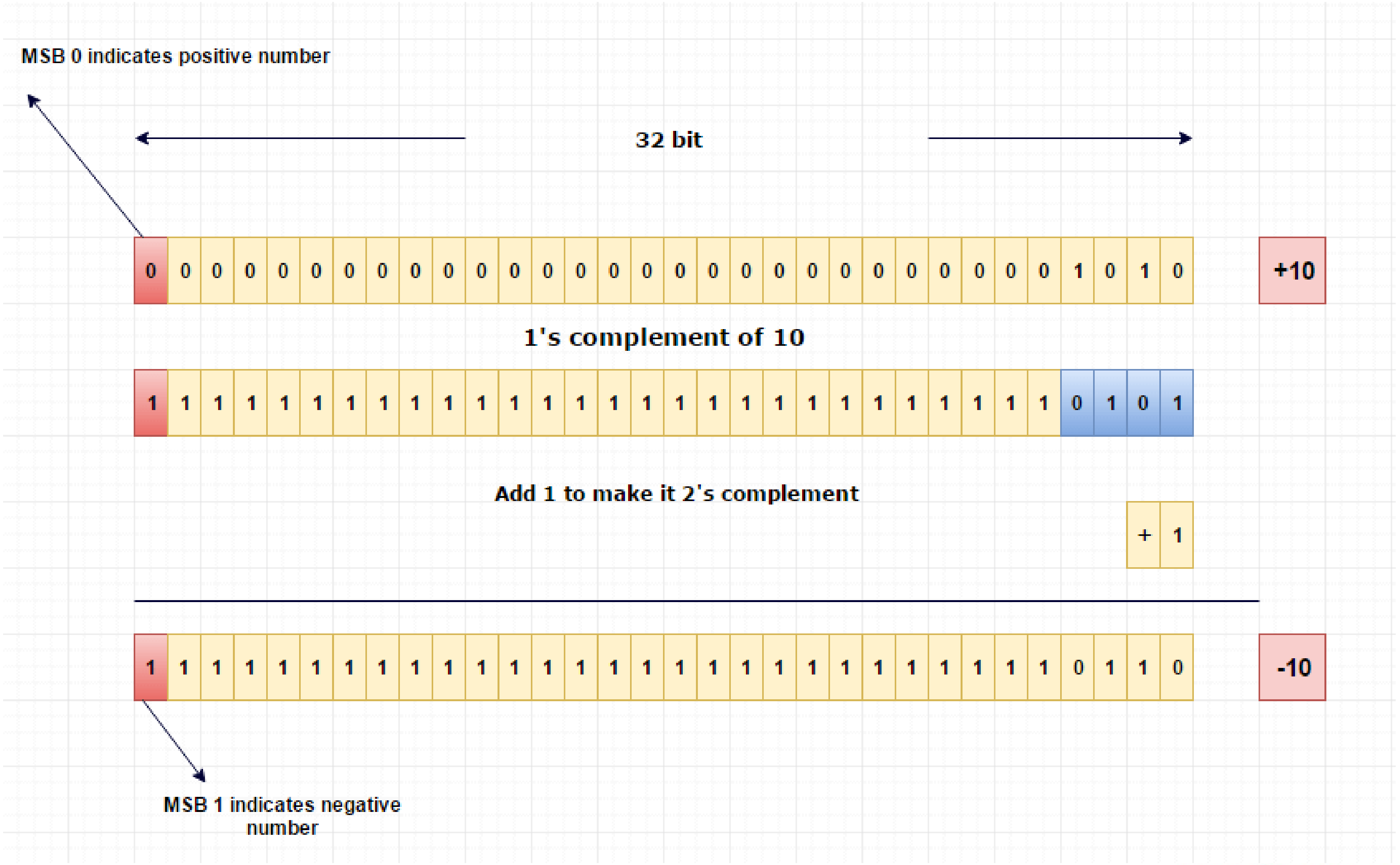


stdout

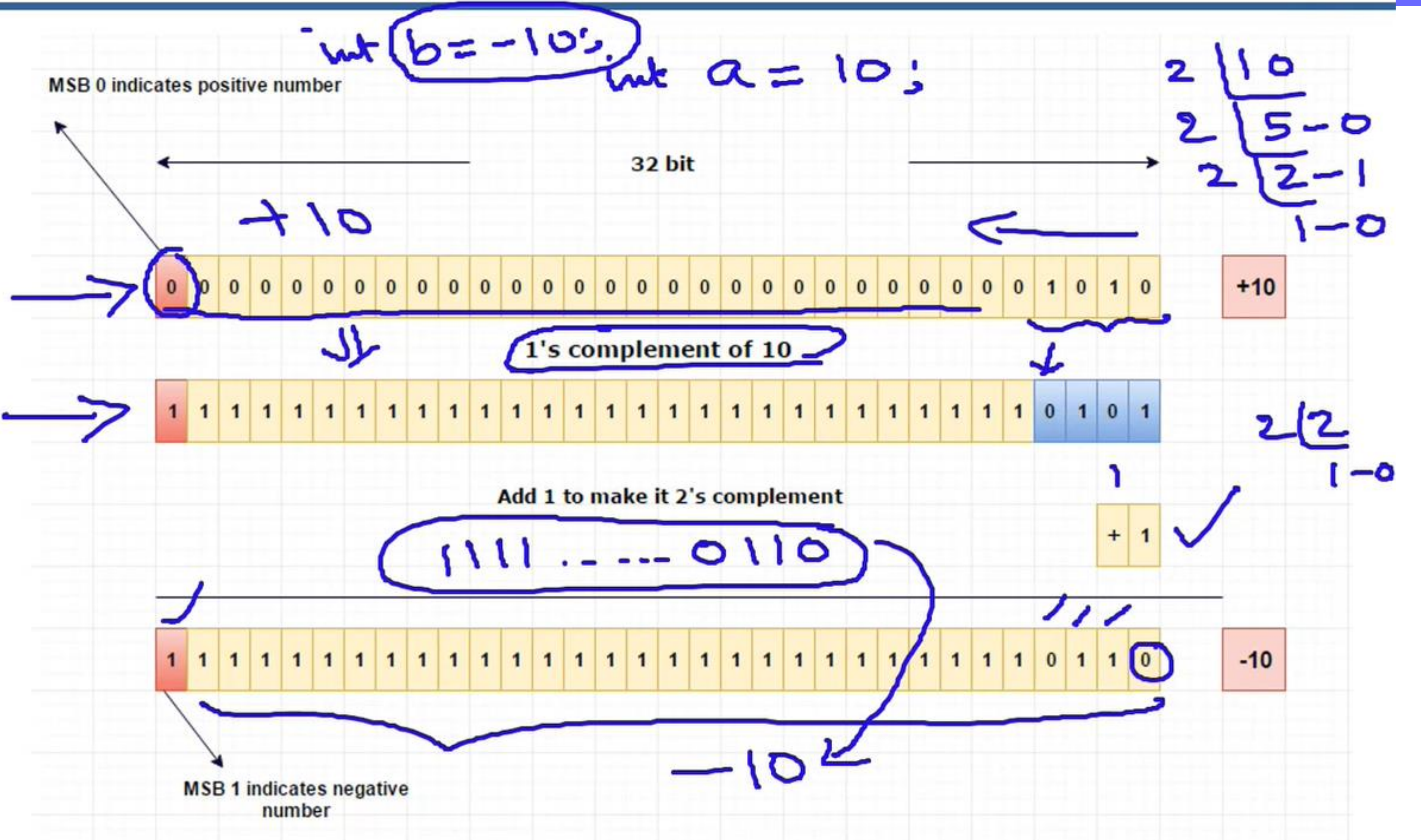
```
The output of the Bitwise complement operator ~a is -61
```

https://ideone.com/9AgdS8

Binary representation of Negative number



Binary representation of Negative number



$$\underbrace{\boxed{1}\ 1100011}$$

$a \rightarrow 60$ $\quad$ $\underbrace{0 \quad 000\boxed{0}}\ 00111100$

$\sim a$? $\boxed{-61}$ ✓ $\quad$ $\sim a \rightarrow \boxed{1}\ 1100011$

$a \rightarrow \boxed{0}\ 00111100$

$\boxed{\sim a \rightarrow \boxed{0}\ \vdots\ 11000011}$ ✓

$1^s\text{comp} \rightarrow \overset{0}{\boxed{1}}\ \vdots\ 00111100$ ✓

$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad 1$

$-(60+1)$

$\hookrightarrow -61$ $\quad$ $2^s\text{comp} \rightarrow \overset{0}{\boxed{1}}\ \vdots\ 00111101$

$\quad\quad\quad \rightarrow \boxed{-61}$

$$\begin{array}{r} 1 \\ 4 \\ 8 \\ 16 \\ 32 \\ \hline 61 \end{array}$$

$$\boxed{\sim a = -(a+1)}$$

Find the binary of $\boxed{-30}$

S1: Find the binary of 30

$$
\begin{array}{r|l}
2 & 30 \\
2 & 15 - 0 \\
2 & 7 - 1 \\
2 & 3 - 1 \\
 & 1 - 1
\end{array}
$$

S2

$$
\begin{array}{rll}
 & \boxed{0} & 0\,0\,0\,1\,1\,1\,1\,0 \\
1^s c & \boxed{1} & 1\,1\,1\,0\,0\,0\,0\,1 \\
 & & 1 \\
\hline
2^s c & \boxed{1} & 1\,1\,1\,0\,0\,0\,1\,0 \;\checkmark \\
\hline
\end{array}
$$

$\left.\right\}$ S3 binary addition

$$
\begin{array}{r|l}
2 & 2 \\
 & 1 - 0
\end{array}
$$

$$
\begin{array}{r}
{}^{1}\,29 \\
72 \\
\hline
101 \\
\hline
\end{array}
$$

$$\left[\underbrace{1\cdots\cdots.1111111}_{24\text{ bits}}\;\underbrace{11100010}_{8\text{ bits}}\right] = -30$$

- Given $y < 0$, and its binary equivalent is 11010011

✓

$\boxed{1}$ ⋮ 1 1 0 1 0 0 1 1

$1^s c$ 0 0 0 1 0 1 1 0 0

$+$ 1

$2^s c$ 0 ⋮ 0 0 1 0 1 1 0 1

$1 \times 2^0 = 1$

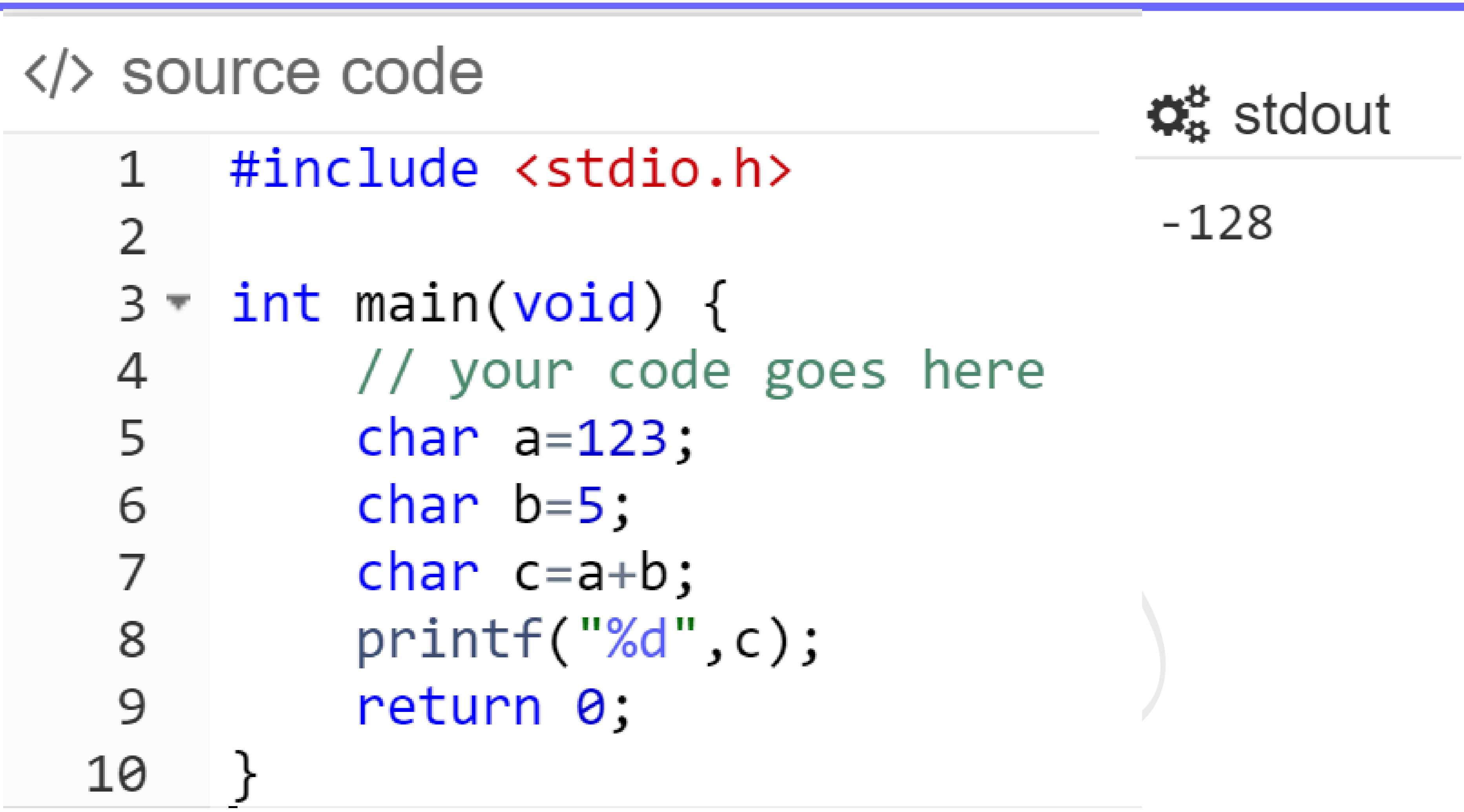
$0 \times 2^1 = 0$

$1 \times 2^2 = 4$

$1 \times 2^3 = 8$

$1 \times 2^5 = 32$

-45


</> source code

```c
#include <stdio.h>

int main(void) {
	// your code goes here
	char a=123;
	char b=5;
	char c=a+b;
	printf("%d",c);
	return 0;
}
```

stdout

```
-128
```

https://ideone.com/a1cYci

char c = a+b;

$$
\begin{array}{rl}
123 \rightarrow & 01111011 \ \checkmark \\
5 \rightarrow & 00000101 \quad + \text{ binary addition} \\
\hline
 & \rightarrow \boxed{1}\underbrace{0000000}
\end{array}
$$

char

$\underbrace{-128 \text{ to } \boxed{127}}$

to $\boxed{0}\underbrace{1111111}$

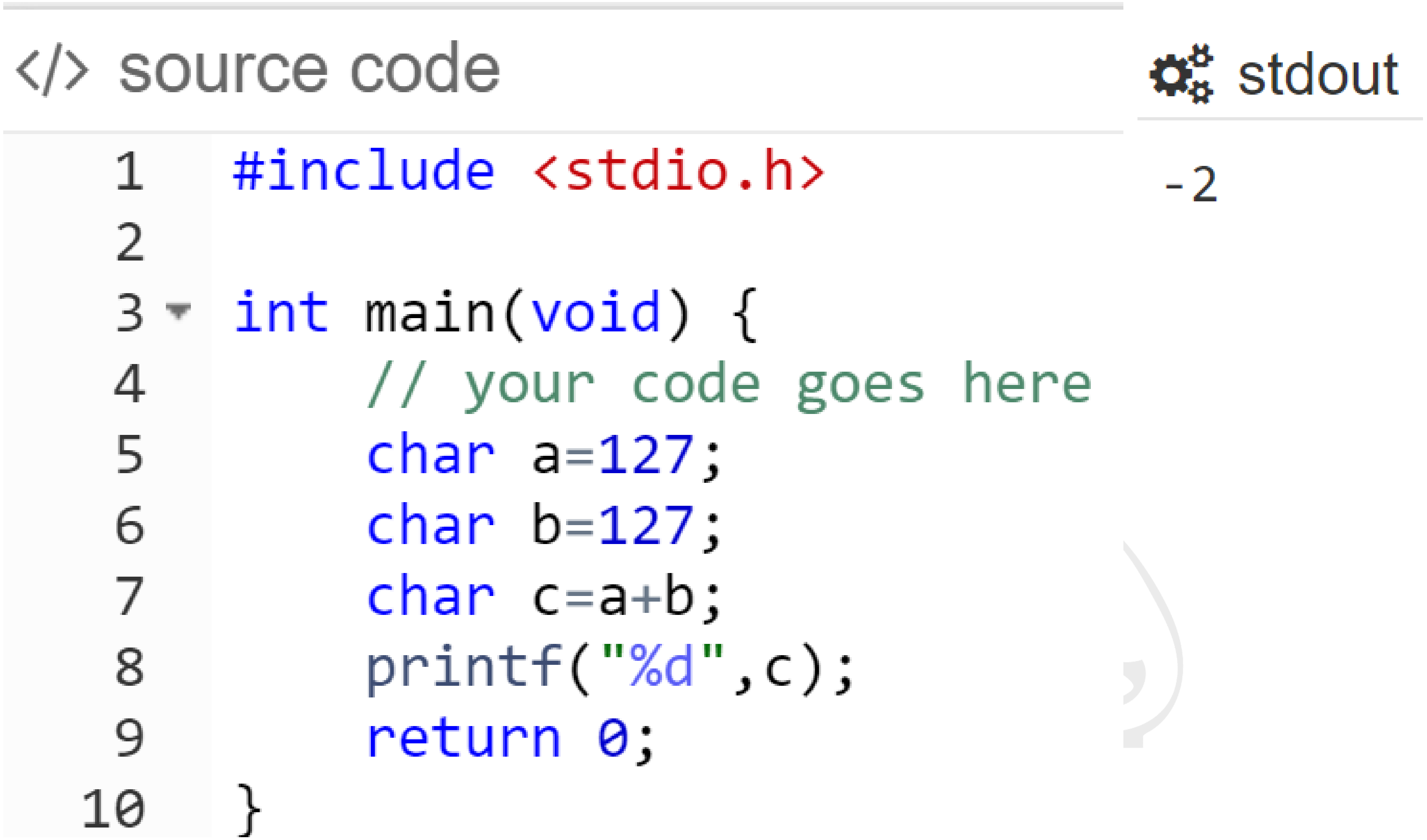
MSB

$$
\begin{array}{rl}
\rightarrow 0000\ldots & \boxed{0}10000000 \\
\hline
1^s c & \boxed{1}01111111 \\
 & \qquad\qquad\quad 1 \\
\hline
1111\ \ 2^s c\ 111 & \boxed{1}\underbrace{10000000}
\end{array}
$$

sign

$\boxed{-128}$ ✓

-30


</> source code

```c
#include <stdio.h>

int main(void) {
	// your code goes here
	char a=127;
	char b=127;
	char c=a+b;
	printf("%d",c);
	return 0;
}
```

stdout

```
-2
```

https://ideone.com/R2tA63

byte

$$\left\{\begin{array}{l} 127 \rightarrow 0\,1\,1\,1\,1\,1\,1\,1 \ \checkmark \\ 127 \rightarrow 0\,1\,1\,1\,1\,1\,1\,1 \end{array}\right.$$

$$\underbrace{\boxed{1}\,1\,1\,1\,1\,1\,1\,0}$$

$y < 0$ and

y ?

$\boxed{-2}$

$\boxed{-45}$

$-128 \ -127 \ . \ . \ 0 \ 1 \ 2 \ . . \ 127$

$7, \boxed{128}$

$0\,\underbrace{1111 - \cdot 1}$

$$\left.\begin{array}{ll} & \boxed{0}\,1\,1\,1\,1\,1\,1\,0 \\ 1\text{'s c} & 1\,0\,0\,0\,0\,0\,0\,1 \\ & 1 \\ \hline 2\text{'s c} & \boxed{1}\,0\,0\,0\,0\,0\,1\,0 \end{array}\right\}$$

$$\boxed{-2} \ \checkmark$$

Thank You!