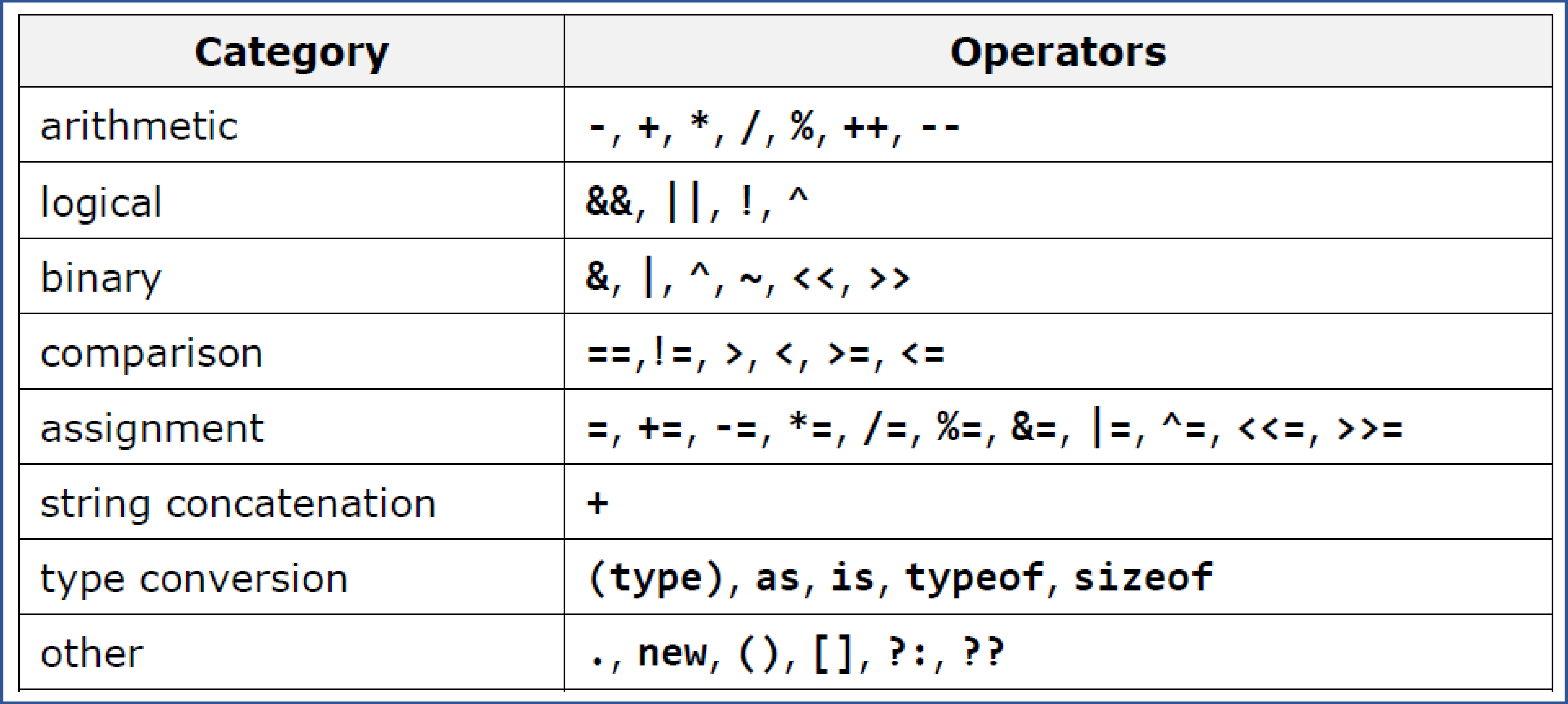
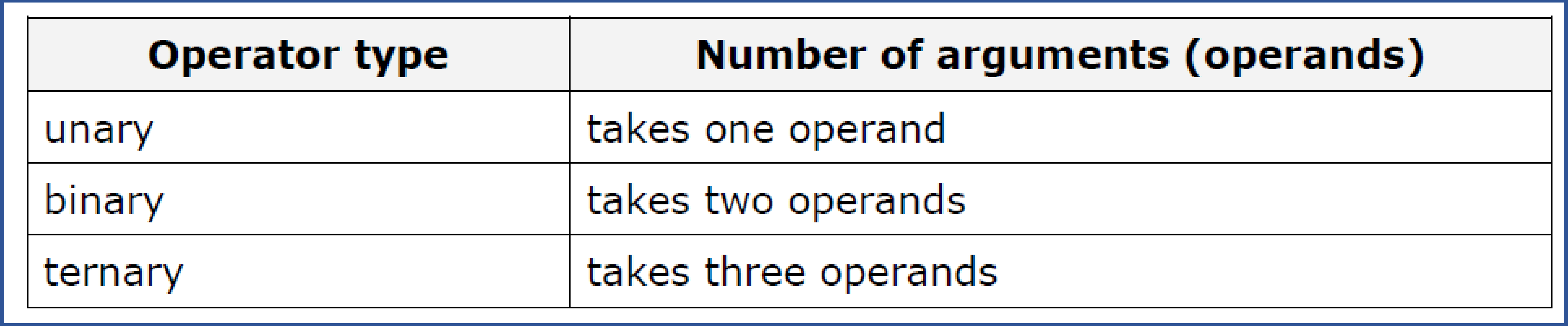
Operators and Expressions

Operator Categories



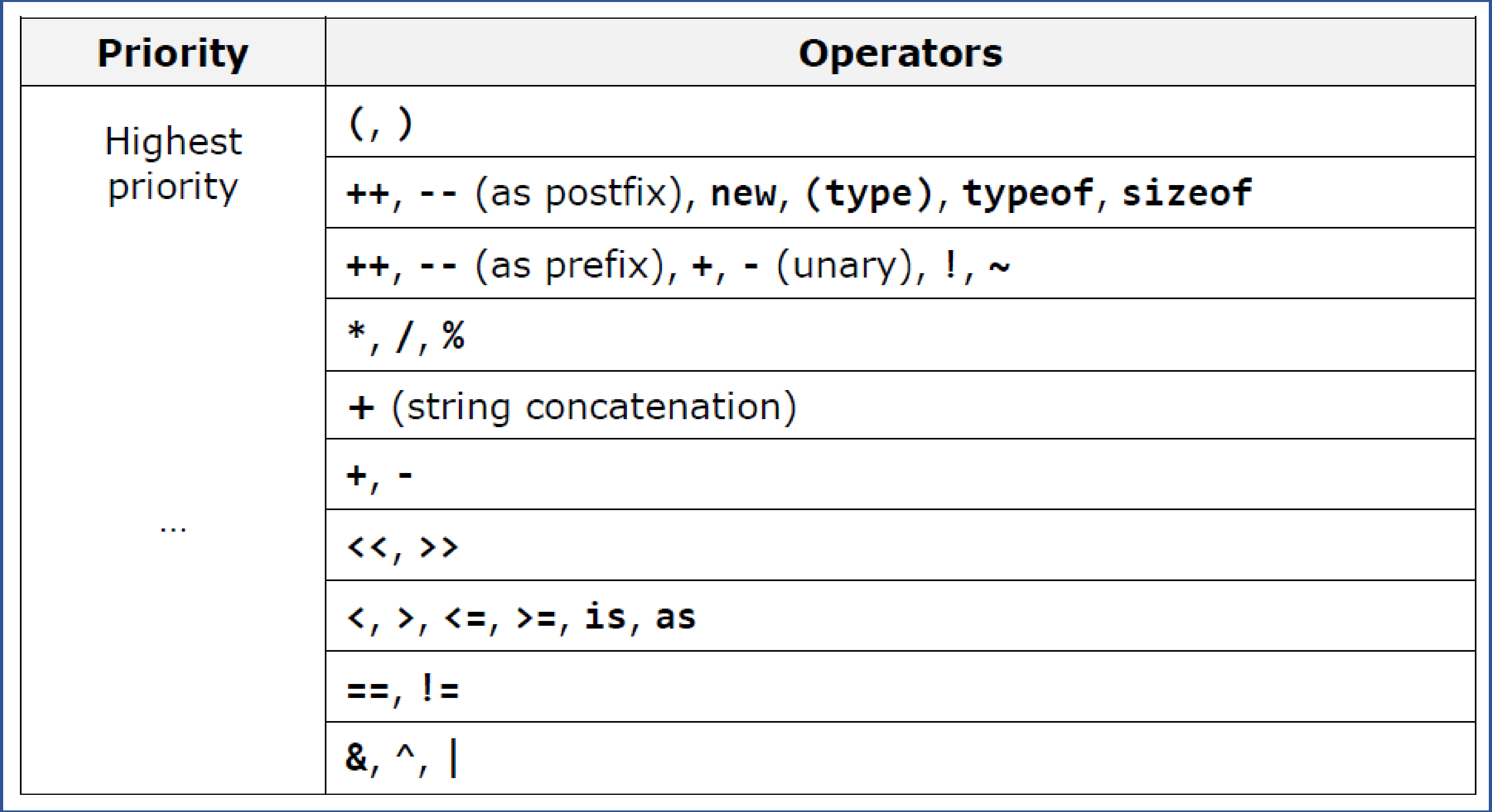
Category	Operators
arithmetic	-, +, *, /, %, ++, --
logical	&&, ||, !, ^
binary	&, |, ^, ~, <<, >>
comparison	==,!=, >, <, >=, <=
assignment	=, +=, -=, *=, /=, %=, &=, |=, ^=, <<=, >>=
string concatenation	+
type conversion	(type), as, is, typeof, sizeof
other	., new, (), [], ?:, ??

Types by Number of Arguments



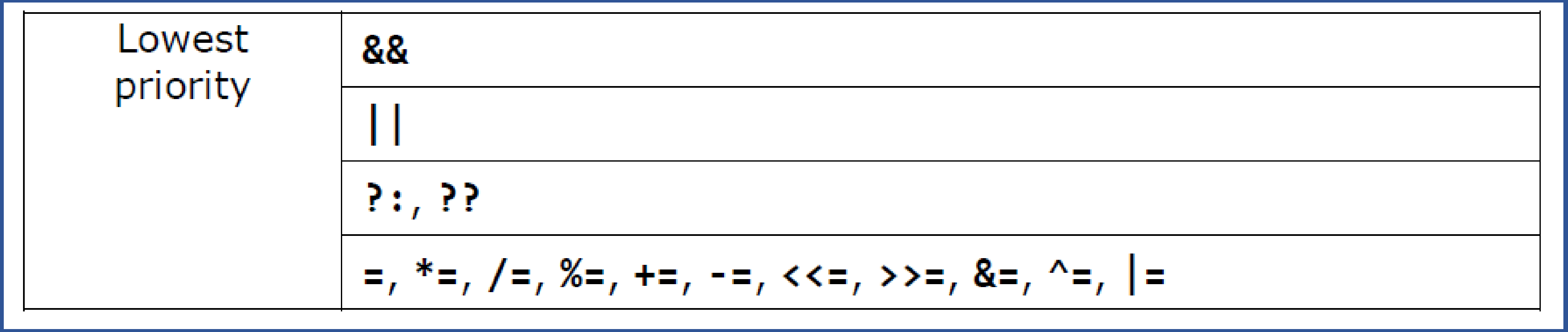
Operator type	Number of arguments (operands)
unary	takes one operand
binary	takes two operands
ternary	takes three operands

Operator Precedence



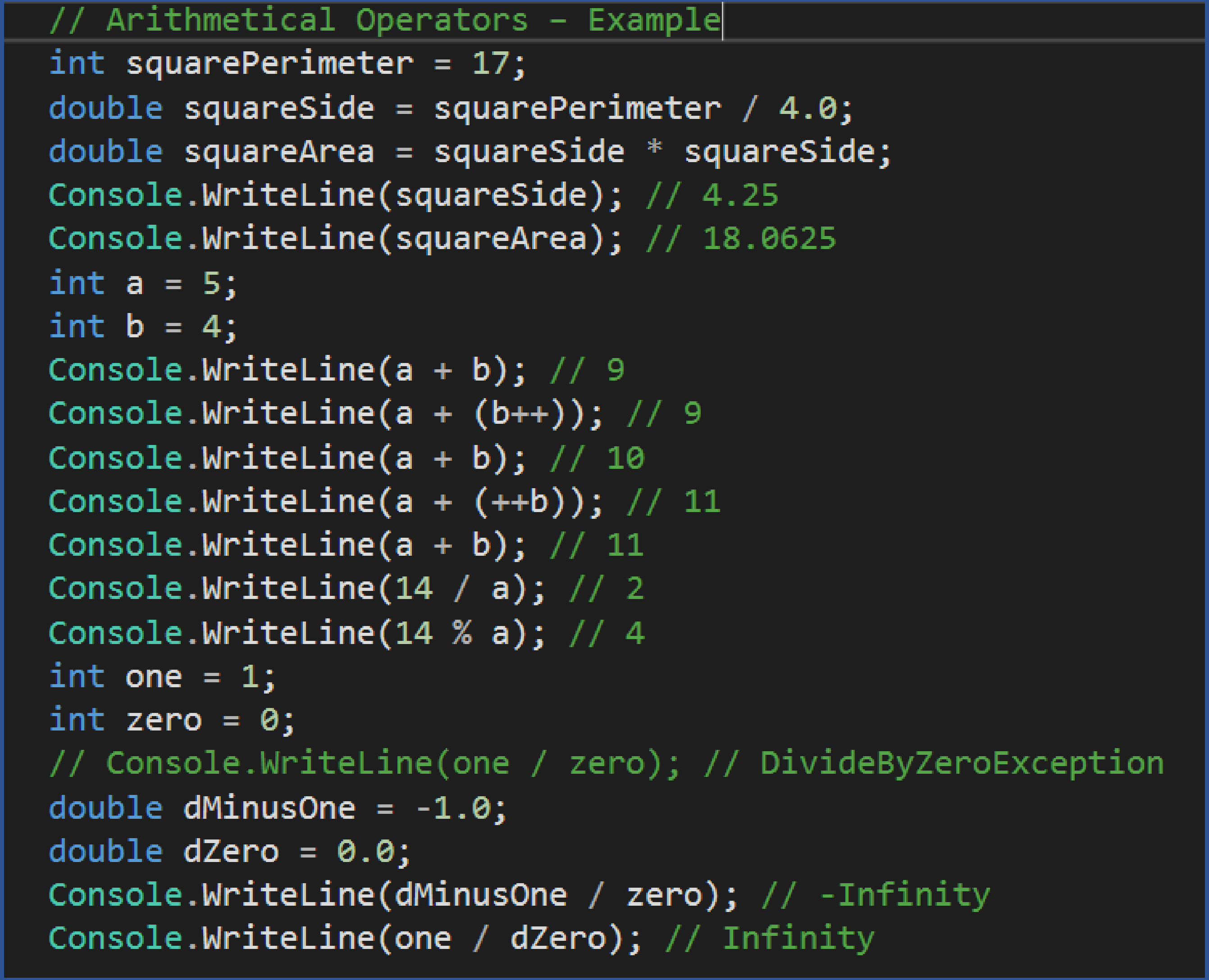
Priority	Operators
Highest priority	**(**, **)**
	++, **--** (as postfix), **new**, **(type)**, **typeof**, **sizeof**
	++, **--** (as prefix), **+**, **-** (unary), **!**, **~**
	*****, **/**, **%**
	+ (string concatenation)
	+, **-**
…	**<<**, **>>**
	<, **>**, **<=**, **>=**, **is**, **as**
	==, **!=**
	&, **^**, **|**

Operator Precedence



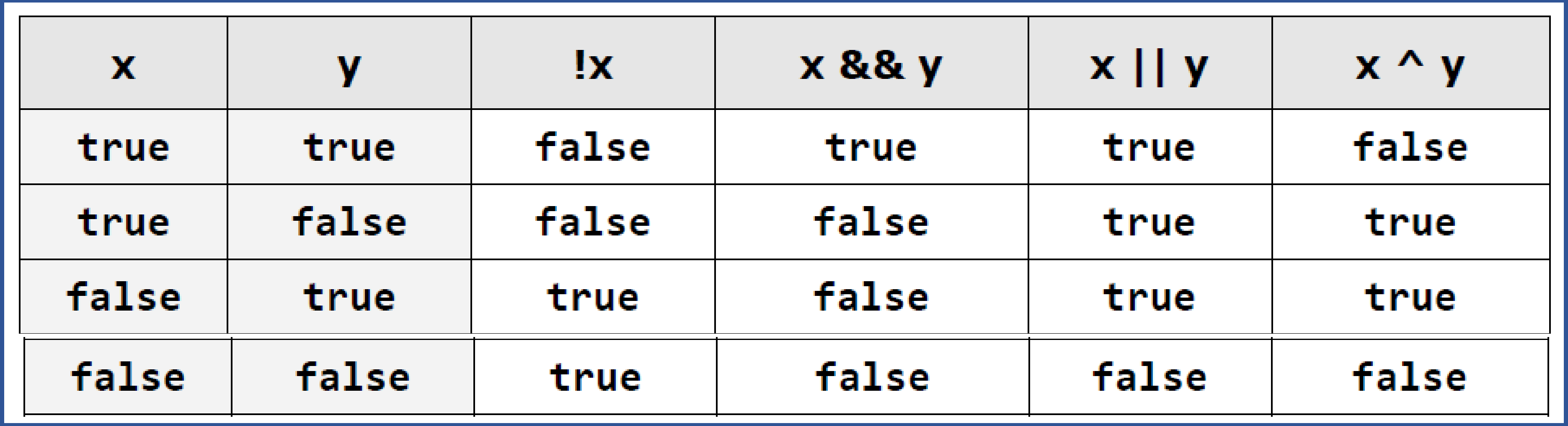
Lowest priority	&&
	||
	?:, ??
	=, *=, /=, %=, +=, -=, <<=, >>=, &=, ^=, |=

Arithmetical Operators – Example

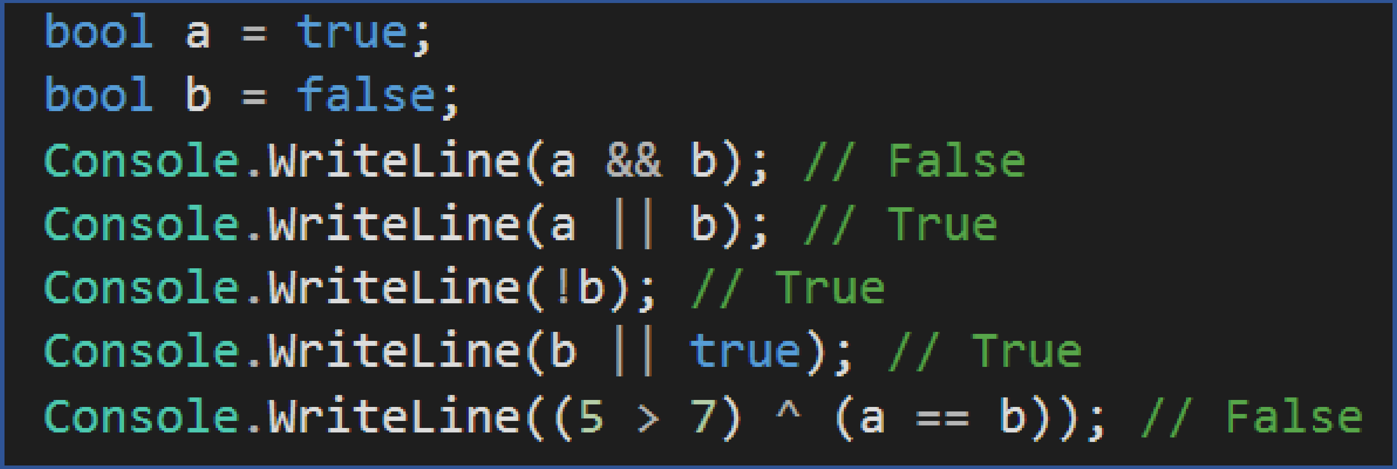


```
// Arithmetical Operators - Example
int squarePerimeter = 17;
double squareSide = squarePerimeter / 4.0;
double squareArea = squareSide * squareSide;
Console.WriteLine(squareSide); // 4.25
Console.WriteLine(squareArea); // 18.0625
int a = 5;
int b = 4;
Console.WriteLine(a + b); // 9
Console.WriteLine(a + (b++)); // 9
Console.WriteLine(a + b); // 10
Console.WriteLine(a + (++b)); // 11
Console.WriteLine(a + b); // 11
Console.WriteLine(14 / a); // 2
Console.WriteLine(14 % a); // 4
int one = 1;
int zero = 0;
// Console.WriteLine(one / zero); // DivideByZeroException
double dMinusOne = -1.0;
double dZero = 0.0;
Console.WriteLine(dMinusOne / zero); // -Infinity
Console.WriteLine(one / dZero); // Infinity
```

Logical Operators

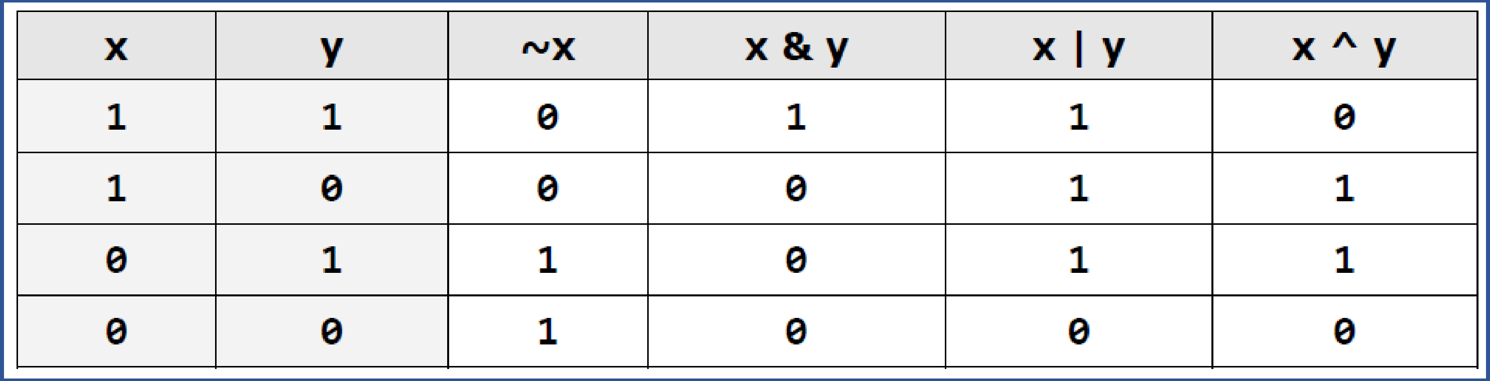


x	y	!x	x && y	x || y	x ^ y
true	true	false	true	true	false
true	false	false	false	true	true
false	true	true	false	true	true
false	false	true	false	false	false

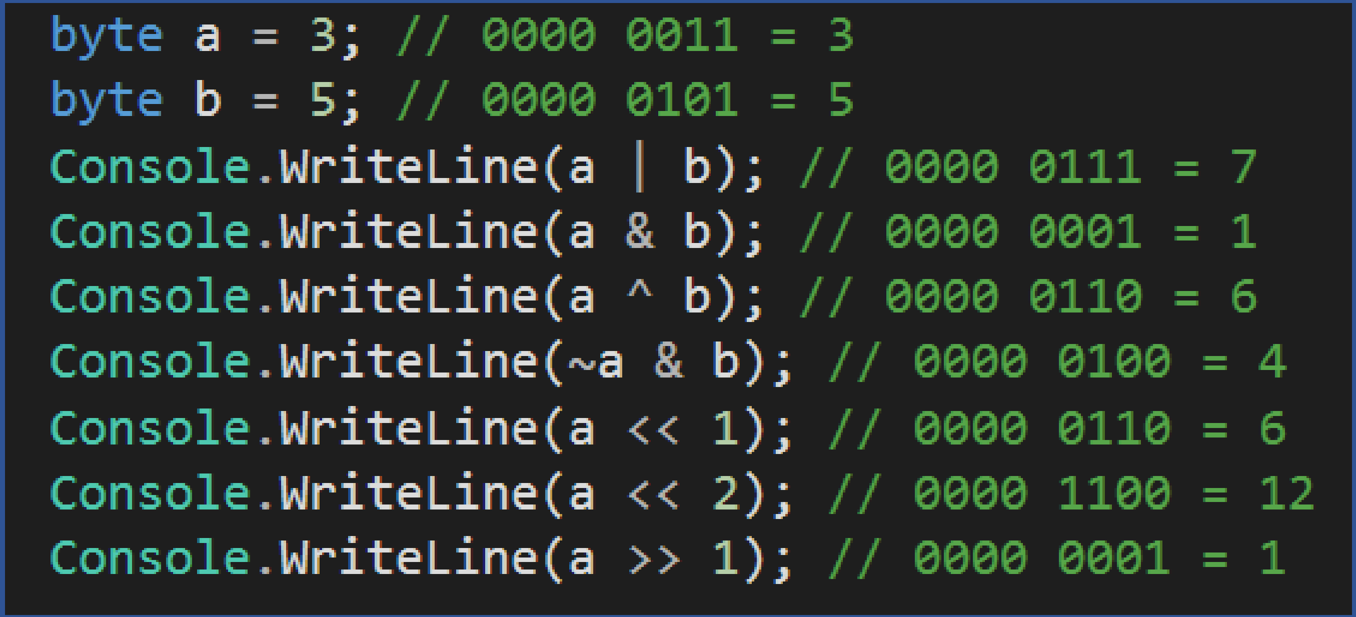


```
bool a = true;
bool b = false;
Console.WriteLine(a && b); // False
Console.WriteLine(a || b); // True
Console.WriteLine(!b); // True
Console.WriteLine(b || true); // True
Console.WriteLine((5 > 7) ^ (a == b)); // False
```

Bitwise Operators

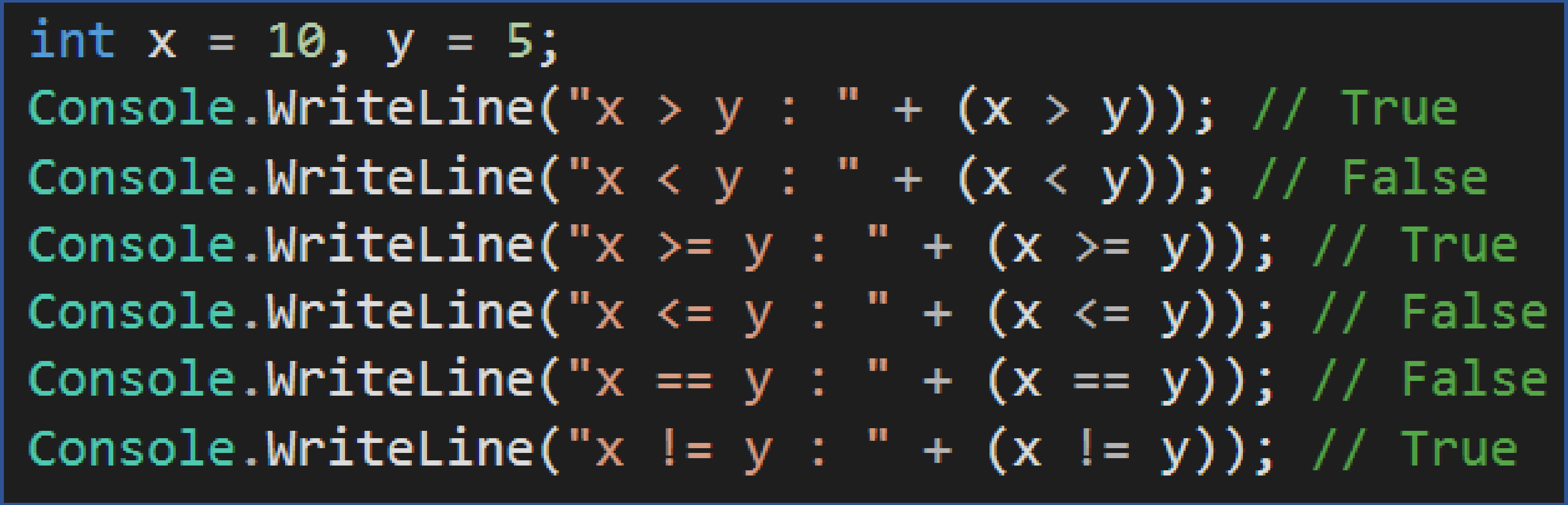


x	y	~x	x & y	x | y	x ^ y
1	1	0	1	1	0
1	0	0	0	1	1
0	1	1	0	1	1
0	0	1	0	0	0



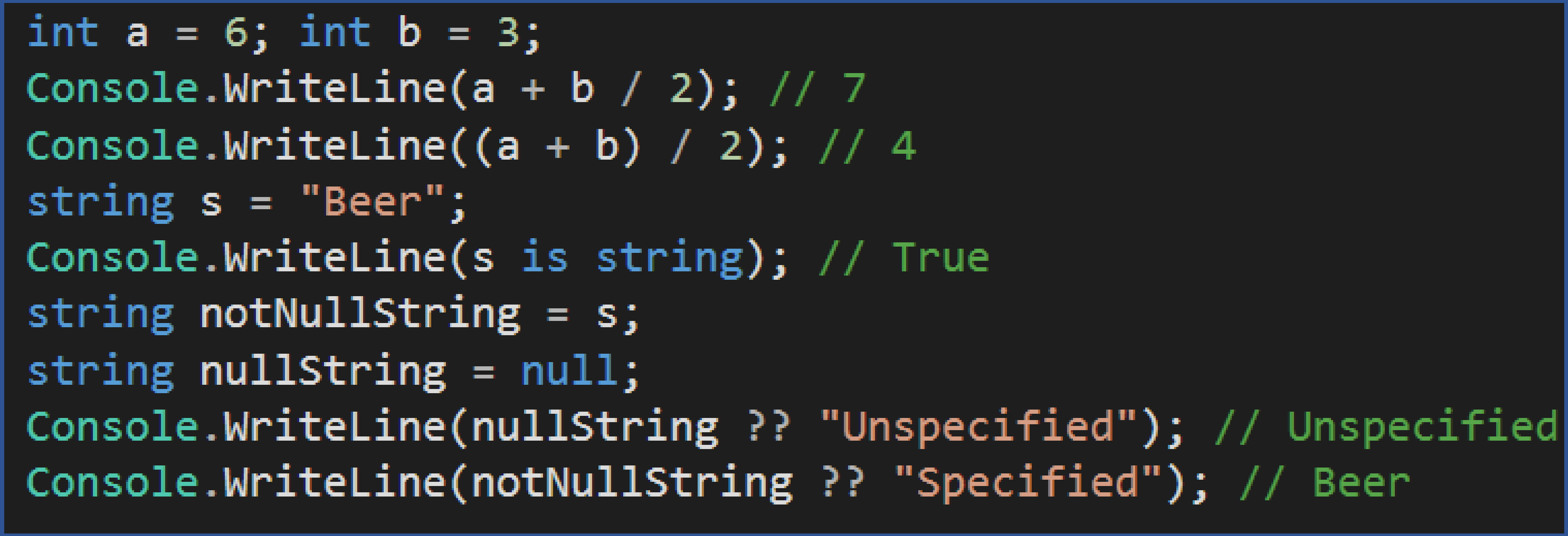
```
byte a = 3; // 0000 0011 = 3
byte b = 5; // 0000 0101 = 5
Console.WriteLine(a | b); // 0000 0111 = 7
Console.WriteLine(a & b); // 0000 0001 = 1
Console.WriteLine(a ^ b); // 0000 0110 = 6
Console.WriteLine(~a & b); // 0000 0100 = 4
Console.WriteLine(a << 1); // 0000 0110 = 6
Console.WriteLine(a << 2); // 0000 1100 = 12
Console.WriteLine(a >> 1); // 0000 0001 = 1
```

Comparison Operators



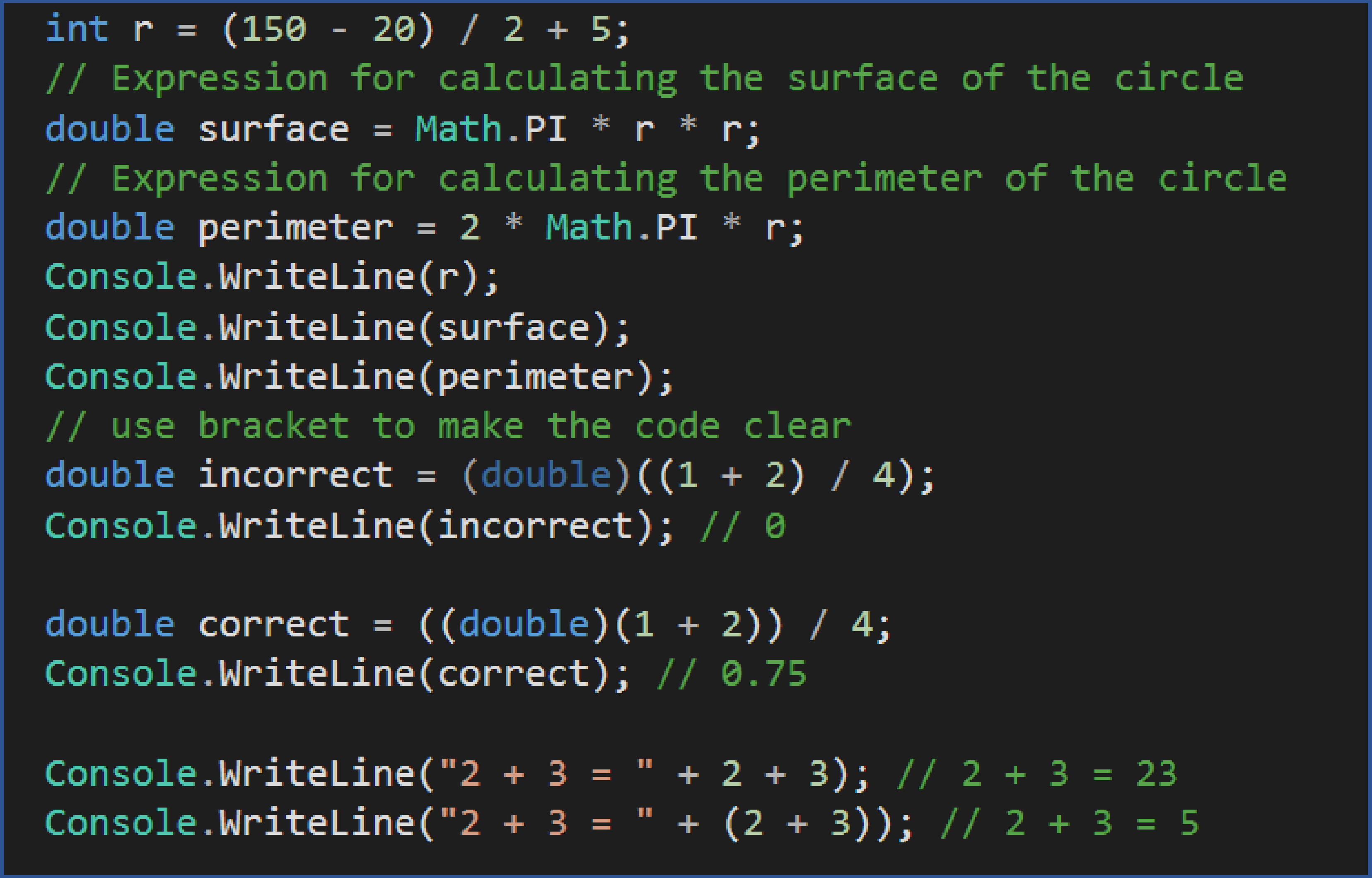
```
int x = 10, y = 5;
Console.WriteLine("x > y : " + (x > y)); // True
Console.WriteLine("x < y : " + (x < y)); // False
Console.WriteLine("x >= y : " + (x >= y)); // True
Console.WriteLine("x <= y : " + (x <= y)); // False
Console.WriteLine("x == y : " + (x == y)); // False
Console.WriteLine("x != y : " + (x != y)); // True
```

Other Operator



```
int a = 6; int b = 3;
Console.WriteLine(a + b / 2); // 7
Console.WriteLine((a + b) / 2); // 4
string s = "Beer";
Console.WriteLine(s is string); // True
string notNullString = s;
string nullString = null;
Console.WriteLine(nullString ?? "Unspecified"); // Unspecified
Console.WriteLine(notNullString ?? "Specified"); // Beer
```

Expression



```
int r = (150 - 20) / 2 + 5;
// Expression for calculating the surface of the circle
double surface = Math.PI * r * r;
// Expression for calculating the perimeter of the circle
double perimeter = 2 * Math.PI * r;
Console.WriteLine(r);
Console.WriteLine(surface);
Console.WriteLine(perimeter);
// use bracket to make the code clear
double incorrect = (double)((1 + 2) / 4);
Console.WriteLine(incorrect); // 0

double correct = ((double)(1 + 2)) / 4;
Console.WriteLine(correct); // 0.75

Console.WriteLine("2 + 3 = " + 2 + 3); // 2 + 3 = 23
Console.WriteLine("2 + 3 = " + (2 + 3)); // 2 + 3 = 5
```

Exercises

1. Write an expression that checks whether an integer is odd or even.

2. Write a Boolean expression that checks whether a given integer is divisible by both 5 and 7, without a remainder.

3. Write an expression that looks for a given integer if its third digit (right to left) is 7.

4. Write an expression that checks whether the third bit in a given integer is 1 or 0.

5. Write an expression that calculates the area of a trapezoid by given sides a, b and height h.

6. Write a program that prints on the console the perimeter and the area of a rectangle by given side and height entered by the user.

7. The gravitational field of the Moon is approximately 17% of that on the Earth. Write a program that calculates the weight of a man on the moon by a given weight on the Earth.

8. Write an expression that checks for a given point {x, y} if it is within the circle K({0, 0}, R=5). Explanation: the point {0, 0} is the center of the circle and 5 is the radius.

9. Write an expression that checks for given point {x, y} if it is within the circle K({0, 0}, R=5) and out of the rectangle [{-1, 1}, {5, 5}]. Clarification: for the rectangle the lower left and the upper right corners are given.

10. Write a program that takes as input a four-digit number in format abcd (e.g. 2011) and performs the following actions:

- Calculates the sum of the digits (in our example 2+0+1+1 = 4).

- Prints on the console the number in reversed order: dcba (in our example 1102).

- Puts the last digit in the first position: dabc (in our example 1201).

- Exchanges the second and the third digits: acbd (in our example 2101).

11. We are given a number n and a position p. Write a sequence of operations that prints the value of the bit on the position p in the number (0 or 1). Example: n=35, p=5 -> 1. Another example: n=35, p=6 -> 0.

12. Write a Boolean expression that checks if the bit on position p in the integer v has the value 1. Example v=5, p=1 -> false.

13. We are given the number n, the value v (v = 0 or 1) and the position p. write a sequence of operations that changes the value of n, so the bit on the position p has the value of v. Example: n=35, p=5, v=0 -> n=3. Another example: n=35, p=2, v=1 -> n=39.

14. Write a program that checks if a given number n (1 < n < 100) is a prime number (i.e. it is divisible without remainder only to itself and 1).

15. * Write a program that exchanges the values of the bits on positions 3, 4 and 5 with bits on positions 24, 25 and 26 of a given 32-bit unsigned integer.

16. * Write a program that exchanges bits {p, p+1, …, p+k-1} with bits {q, q+1, …, q+k-1} of a given 32-bit unsigned integer.