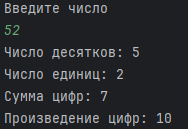
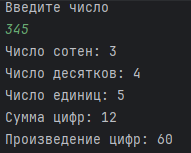
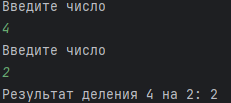
1) fun main() {  
 *println*("Введите число")  
 val a = *readln*()!!.*toInt*()  
 val b = a / 10  
 val c = a % 10  
 val d = b + c  
 val f = b \* c  
 *println*("Число десятков: $b")  
 *println*("Число единиц: $c")  
 *println*("Сумма цифр: $d")  
 *println*("Произведение цифр: $f")  
}



2) fun main() {  
 *println*("Введите число")  
 val a = *readln*()!!.*toInt*()  
 val hundreds = a / 100  
 val tens = (a / 10) % 10  
 val units = a % 10  
 val sum = hundreds + tens + units  
 val product = hundreds \* tens \* units  
 *println*("Число сотен: $hundreds")  
 *println*("Число десятков: $tens")  
 *println*("Число единиц: $units")  
 *println*("Сумма цифр: $sum")  
 *println*("Произведение цифр: $product")  
}



3) fun main() {  
 *println*("Введите число")  
 val a = *readln*()!!.*toInt*()  
 *println*("Введите число")  
 val b = *readln*()!!.*toInt*()  
 if (b != 0) {  
 val result = a / b  
 *println*("Результат деления $a на $b: $result")  
 } else {  
 *println*("Деление на ноль невозможно.")  
 }  
 }



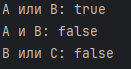
4) fun main() {  
 val base = 3.0  
 val exponent = 4  
 val result = Math.pow(base, exponent.toDouble())  
 *println*("$base в степени $exponent равно $result")  
}



5) fun main() {  
 val number = 42.0  
 if (number >= 0) {  
 val root = Math.sqrt(number)  
 *println*("Корень числа $number равен $root")  
 } else {  
 *println*("Корень отрицательного числа не существует.")  
 }  
}



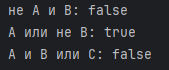
1) fun main() {  
 val A = true  
 val B = false  
 val C = false  
 *println*("A или B: ${A || B}")  
 *println*("A и B: ${A && B}")  
 *println*("B или C: ${B || C}")  
}



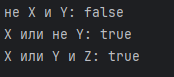
2) fun main() {  
 val X = false  
 val Y = true  
 val Z = false  
 *println*("X или Z: ${X || Z}")  
 *println*("X и Y: ${X && Y}")  
 *println*("X и Z: ${X && Z}")  
}



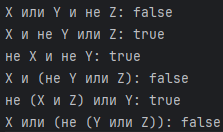
3) fun main() {  
 val A = true  
 val B = false  
 val C = false  
 *println*("не A и B: ${!A && B}")  
 *println*("A или не B: ${A || !B}")  
 *println*("A и B или C: ${A && B || C}")  
}



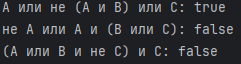
4) fun main() {  
 val X = true  
 val Y = true  
 val Z = false  
 *println*("не X и Y: ${!X && Y}")  
 *println*("X или не Y: ${X || !Y}")  
 *println*("X или Y и Z: ${X || (Y && Z)}")  
}



5) fun main() {  
 val X = false  
 val Y = false  
 val Z = true  
 *println*("X или Y и не Z: ${X || (Y && !Z)}")  
 *println*("X и не Y или Z: ${X && !Y || Z}")  
 *println*("не X и не Y: ${!X && !Y}")  
 *println*("X и (не Y или Z): ${X && (!Y || Z)}")  
 *println*("не (X и Z) или Y: ${!(X && Z) || Y}")  
 *println*("X или (не (Y или Z)): ${X || !(Y || Z)}")  
}



6) fun main() {  
 val A = true  
 val B = false  
 val C = false  
 *println*("A или не (A и B) или C: ${A || !(A && B) || C}")  
 *println*("не A или A и (B или C): ${!A || (A && (B || C))}")  
 *println*("(A или B и не C) и C: ${(A || (B && !C)) && C}")  
}



7) fun main() {  
 val A = true  
 val B = false  
 val C = false  
 val resultA = A || !(A && B) || C  
 val resultB = !A || (A && (B || C))  
 val resultC = (A || (B && !C)) && C  
 *println*("Результат a) A ∨ ¬(A ∧ B) ∨ C: $resultA")  
 *println*("Результат b) ¬A ∨ (A ∧ (B ∨ C)): $resultB")  
 *println*("Результат c) (A ∨ (B ∧ ¬C)) ∧ C: $resultC")  
}

