# Exercises: Operators Expressions and Statements

This document defines in-class exercises from the ["C# Basics" Course @ Software University](http://softuni.bg/courses/csharp-basics/).

### Arithmetic Operators

## Average

Write a program that find the **average** of the **sum of 3** numbers.

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **c** | **Average** |
| 45 | 41 | 20 | 35.33333 |
| 22 | 52 | 60 | 44.66667 |

## Trapezoid

Write a program that finds the **area** of a trapezoid, given the base sides **a**, **b** and height **h**.

|  |  |  |  |
| --- | --- | --- | --- |
| **a** | **b** | **h** | **Area** |
| 5 | 2 | 4 | 14 |
| 8.5 | 4.4 | 2 | 12.9 |

## Last Digit

Write a program that prints the last digit of a number **n**.

|  |  |
| --- | --- |
| **n** | **Result** |
| 21 | 1 |
| 139 | 9 |
| 4 | 4 |

## N-th Digit

Write a program that prints the **n**-th digit of a number (from right to left). If no such digit exists, print a dash "**-**".

|  |  |  |
| --- | --- | --- |
| **Number** | **n** | **Result** |
| 2174 | 3 | 1 |
| 169 | 2 | 6 |
| 46 | 4 | - |

### Logical Operators

## Big and Odd

Write a program that that prints if the number is both **greater than 20** and **odd**.

|  |  |
| --- | --- |
| **n** | **Result** |
| 63 | true |
| 17 | false |
| 22 | false |
| 23 | true |
| 20 | false |

## Pure Divisor

Write a program that prints if a number **divides** by either 9, 11 or 13 **without remainder**.

|  |  |
| --- | --- |
| **n** | **Result** |
| 121 | true |
| 1263 | false |
| 26 | true |
| 23 | false |
| 81 | true |
| 1287 | true |

### Bitwise Operators

## First Bit

Write a program that prints the bit at **position 1** of a number.

|  |  |
| --- | --- |
| **n** | **Result** |
| 2 | 1 |
| 51 | 1 |
| 13 | 0 |
| 24 | 0 |

## p-th Bit

Write a program that prints the bit at position **p** of a number.

|  |  |  |
| --- | --- | --- |
| **n** | **p** | **Result** |
| 2145 | 5 | 1 |
| 512 | 0 | 0 |
| 111 | 8 | 0 |
| 255 | 7 | 1 |

## Bit Destroyer

Write a program that sets all bits at **position** **p** to **0**. Print the resulting number.

|  |  |  |
| --- | --- | --- |
| **n** | **p** | **Result** |
| 1313 | 5 | 1281 |
| 231 | 2 | 227 |
| 111 | 6 | 47 |

## \* Tri-bit Switch

Write a program that inverts the **3 bits** from position **p** to the left with their opposites (e.g. **111** -> **000**, **101** -> **010**). Print the resulting number on the console.

|  |  |  |  |
| --- | --- | --- | --- |
| **n** | **p** | **Result** | **Details** |
| 3652421623 | 7 | 3652420727 | **1101 1001 1011 0011 1000 0111 1111 0111**  **1101 1001 1011 0011 1000 0100 0111 0111** |
| 2135693832857285912 | 59 | 2712154585160709400 | **0001 1101 1010 0011 1000 0010 0011 1001 1100 0000 0111 0011 0111 0101 0001 1000**  **0010 0101 1010 0011 1000 0010 0011 1001 1100 0000 0111 0011 0111 0101 0001 1000** |