#### Logic Diagrams Activity

It is useful to have a visual representation of Boolean expressions so that you can draw diagrams of simple Boolean logic gates.

The **Name** column and the **Operation** column are incorrectly aligned to the **Diagram** column, correct them.

|  |  |  |
| --- | --- | --- |
| **Name** | **Diagram** | **Operation** |
| **AND** | **A**  **B**  **Q** | Takes two Boolean values as input and returns ‘true’ if both inputs are ‘true’, or ‘false’ if either input is ‘false’. |
| **OR** | **A**  **B**  **Q** | Takes two Boolean values as input and returns ‘true’ if either input is ‘true’, or ‘false’ if both inputs are ‘false’. |
| **NOT** | **A**  **Q** | Takes a single Boolean input and returns ‘true’ if the input is ‘false’, or ‘false’ if the input is ‘true’. |
| **NAND** | **Q**  **A**  **B** | Taking two Boolean values as input, and returns the inverse of an AND, i.e. ‘true’ if either input is ‘false’, or ‘false’ if both inputs are ‘true’. |
| **NOR** | **Q**  **B**  **A** | Taking two Boolean values as input, and returns ‘true’ if both inputs are ‘false’, or ‘false’ if either input is ‘true’. |
| **XOR** | **Q**  **B**  **A** | Taking two Boolean values as input, and returns ‘true’ if either input is exclusively ‘true’ (i.e. not both). |
| **XNOR** | **Q**  **B**  **A** | Describe its operation:  Taking two Boolean values as input, and returns ‘false’ if either input is exclusively ‘true’ (i.e. not both). Inverse of XOR |

**How have you done: \_\_\_\_\_\_14**

#### Truth Tables

Boolean operations can be represented as truth tables, which show the outputs for every possible combination of inputs. Complete the truth tables.

|  |  |  |  |
| --- | --- | --- | --- |
| **AND** |  |  | **Output** |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 0 |
|  | 1 | 0 | 0 |
|  | 1 | 1 | 1 |

|  |  |  |  |
| --- | --- | --- | --- |
| **OR** |  |  | **Output** |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | 1 |

|  |  |  |
| --- | --- | --- |
| **NOT** |  | **Output** |
|  | 0 | 1 |
|  | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **NAND** |  |  | **Output** |
|  | 0 | 0 | 1 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **NOR** |  |  | **Output** |
|  | 0 | 0 | 1 |
|  | 0 | 1 | 0 |
|  | 1 | 0 | 0 |
|  | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **XOR** |  |  | **Output** |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| **XNOR** |  |  | **Output** |
|  | 0 | 0 | 0 |
|  | 0 | 1 | 1 |
|  | 1 | 0 | 1 |
|  | 1 | 1 | 1 |

**How have you done: \_\_\_\_\_\_5**

**Algebraic Terms**

|  |  |
| --- | --- |
| **INPUT** | **Write out the Term** |
| **AND** | . |
| **OR** | + |
| **NOT** | \_  A |
| **NAND** | \_\_  AB |
| **NOR** | \_\_\_  A+B |
| **XOR** | AOB |
| **XNOR** | \_\_\_  AOB |

**How have you done: \_\_\_\_\_\_7**

**Combinational**

Logic gates can be combined to form more complex logic circuits.

For example, the logic circuit for the Boolean expression (NOT (A AND B)) OR C is as follows:

**B**

**A**

**E**

**D**

**Q**

**C**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  |
|  |  |  |  |  |  |
| 0 | 0 | 0 |  |  |  |
| 0 | 0 | 1 |  |  |  |
| 0 | 1 | 0 |  |  |  |
| 0 | 1 | 1 |  |  |  |
| 1 | 0 | 0 |  |  |  |
| 1 | 0 | 1 |  |  |  |
| 1 | 1 | 0 |  |  |  |
| 1 | 1 | 1 |  |  |  |

**How have you done: \_\_\_\_\_\_3**

**Overall Total: \_\_\_\_\_26**