# Problem 5 – Bit Paths

You are given a **board** of size **8 rows \* 4 columns**. It is initially empty (holds zeroes). A **bit path** starts at certain column at the first row and falls 7 times either **down left** (direction = -1), **down** (direction = 0) or **down right** (direction = +1). Bit paths **invert all visited cells** in the board (turns 0 to 1 and 1 to 0). Bit paths are represented as strings of 8 items, separated by a comma: a **start column** + **7 directions**. For example, the bit path "2,-1,-1,+1,-1,+1,+1,-1" starts from column 2 at the first row, moves to column 1 at the second row, then to column 0 at the third row, etc. The bit paths will never go out of the board. See the below examples to catch the idea.

After the bit paths are processed, the **bits of each board row** are taken as binary number, converted to **decimal**, and then **summed**. Finally, the obtained sum is printed in **binary** and **hexadecimal** form. The below example illustrates the entire process:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input Data** | **Initial Board** | **Process the 1st Path** | **Process the 2nd Path** | **Process the 3rd Path** | **Output** |
| 3  2,-1,-1,+1,-1,+1,+1,-1  0,0,+1,+1,0,-1,0,-1  3,0,0,-1,-1,-1,0,+1 | 0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0  0 0 0 0 | 0 0 **1** 0  0 **1** 0 0  **1** 0 0 0  0 **1** 0 0  **1** 0 0 0  0 **1** 0 0  0 0 **1** 0  0 **1** 0 0 | **1** 0 1 0  **1** 1 0 0  1 **1** 0 0  0 1 **1** 0  1 0 **1** 0  0 **0** 0 0  0 1 **1** 0  1 **1** 0 0 | 1 0 1 **1**  1 1 0 **1**  1 1 0 **1**  0 1 **0** 0  1 **1** 1 0  **1** 0 0 0  **1** 1 1 0  1 **0** 0 0 | Decimal(1011) = 11, Sum = 11  Decimal(1101) = 13, Sum = 24  Decimal(1101) = 13, Sum = 37  Decimal(0100) = 4, Sum = 41  Decimal(1110) = 14, Sum = 55  Decimal(1000) = 8, Sum = 63  Decimal(1110) = 14, Sum = 77  Decimal(1000) = 8, Sum = 85  Binary(85) = **1010101**; Hex(85) = **55** |

## Input

The input data should be read from the console. The first line holds the number of paths **n**. The next **n** lines hold **n paths** (in the above-described format). The input data will always be valid and in the format described.

## Output

Print at the console two lines: the sum of board rows in binary and hexadecimal form.

## Constraints

* The count of paths **n** is an integer in the range [0…100].
* The **start positions** are a positive numbers in the range [0…3].
* The **directions** are one of the following values: **-1**, **0**, **+1**.
* Allowed working time: 0.25 seconds. Allowed memory: 16 MB.

## Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 3  2,-1,-1,+1,-1,+1,+1,-1  0,0,+1,+1,0,-1,0,-1  3,0,0,-1,-1,-1,0,+1 | 1010101  55 |  | 2  2,-1,-1,+1,-1,+1,+1,-1  0,0,+1,+1,0,-1,0,-1 | 1000100  44 |  | 1  2,0,+1,0,0,0,-1,+1 | 1011  B |