# Problem 5 – Bit Builder

The rules of the game are as follows:

You are given a sequence of bits (an **integer** number) and Pesho chooses a position and issues an order in order to manipulate the given bit. If he says **flip,** you have to reverse the value of the bit. For example if the bit’s value is 1, it has to become 0. If Pesho’s order is **remove**, you have to remove the bit from the bit sequence (1 1100 1**1**01 🡪 0 1110 0101). However, if he issues the order **insert** the bit **1** has to be insertedin the wanted position (0 1110 0101 🡪 1 1100 1**1**01). If he issues the order **skip,** you don’t have to do anything with the given bit. Whenever Pesho says **quit**, the game ends.

# Input

The input data should be read from the console. On the first line, you are given an integer number and on each of the next two lines, you have a bit position and an issued order.

The possible orders are as follows: “**flip**”, “**remove**”, “**insert**”, “**skip**”. On the last input line, you are given the order “**quit**”, which means that the game has ended.

The input data will always be valid and in the format described. There is no need to check it explicitly.

# Output

On the only output line print the bits of the number after the end of the game

# Constraints

* The **input number** will be a 32-bit integer in the range [0 … 2 147 483 647].
* The position will be in the range [0 … 31].
* The maximum number of commands will be 30.
* Allowed working time for your program: 0.1 seconds. Allowed memory: 16 MB.

# Examples

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Input** | **Output** | **Explanation** | **Input** | **Output** | **Explanation** |
| 255  0  flip  10  insert  1  remove  8  skip  quit | 638 | 1111 1111  🡪 000 1111 1110  🡪 100 1111 1110  🡪 10 0111 1110  🡪 10 0111 1110 (638) | 11230  9  insert  15  remove  0  remove  2  flip  quit | 11243 | 10 1011 1101 1110  🡪 0101 0111 1101 1110  🡪 101 0111 1101 11107  🡪 10 1011 1110 1111  🡪 10 1011 1110 1011  (11243) |