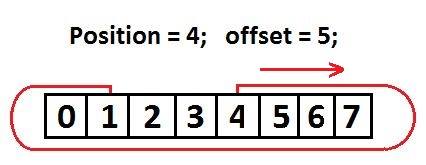
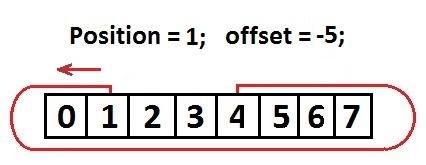
# Problem 2 – Array Slider

Фил не е мил човек. Мислите, че на изпита задачата с масив ще е най-лесна, но той прави всичко по силите си да я направи по-трудна. Също така Фил харесва хора, които сляпо следват заповеди, така че ще ви даде масив с цели числа и поредица от операции, които да изпълните с него.

На първия ред от входа ще ви бъде дадени няколко **неотрицателни цели числа, отделени с празно място (едно или повече)**.

На следващите редове до получаването на командата "**stop**" ще получавате команди, които ще описват какво да правите. Командите ще са във **формата "[отместване] [операция] [операнд]"**. Отместването ще е число, което показва кой елемент ще манипулирате. **Започвате с елемента на позиция 0 и добавяте отместването на всяка стъпка**. Ако получите командата "2 \* 2", това значи, че трябва да работите с елемента на позиция 2 = тоест 0 (първоначалния индекс) + 2 (отместването). Като следваща команда получавате "-2 \* 2", това значи, че трябва да оперирате с елемента на индекс 0 = 2 (настоящия индекс) + -2 (отместването). Ако получите **положително отместване и свършите извън обхвата на масива**, **започнете отначало**. Същото важи и за **отрицателни отмествания; ако се окажете с индекс по-малък от 0, започнете от края на масива**.



Внимавайте кой е настоящият индекс, започвате от 0, но **всяко отместване се отчита спрямо индекса, с който последно сте работили**.

**Операцията** ще е един от следните символи: **'&' (побитово AND), '|' (побитово OR), '^' (побитово XOR), '+' (прибави), '-' (извади), '\*' (умножи), '/' (раздели)**. **Операндът** ще е **неотрицателно цяло число**. Това, което трябва да направите, е: **1) открийте индекса на обработвания елемент**, като следвате правилата по-горе; **2) извършете операцията** с избрания елемент. Смятайте, следвайте дадените команди и, когато сте готови, **отпечатайте масива на конзолата** във формат: **"[arr0, arr1, …, arrN]"**.

**Резултантният масив трябва да съдържа само неотрицателни числа**! Това значи, че когато **изваждате**, ако получите отрицателно число резултатът ви трябва да стане 0. Всички други операции трябва да получават **неотрицателни** числа.

### Вход

* Входната информация трябва да се чете от конзолата.
* Първият ред на входа трябва да съдържа **поредица от цели числа**, отделени с **едно или повече празни места**.
* Следващите редове да съдържат **команди** във формата **"[отместване] [операция] [операнд]"**.
* Като последен ред от входа ще получите **командата "stop"**.
* Входната информация винаги ще е валидна и във описания формат. Няма нужда от специална проверка.

### Изход

* Изходът да се отпечата на конзолата.
* След изпълняване на командите да се отпечата **финалният масив във формат "[arr1, arr2, …]"**. Без отрицателни числа в изхода!

### Ограничения

* **Броят цели числа** в колекцията ще е в диапазона [1 … 50].
* **Броят команди** ще е в диапазона [1 … 20].
* **Всички числа във входния масив** и **операндите** ще са цели числа в диапазона [0 … 231 - 1].
* **Отместването** ще е в диапазона [-231 … 231 - 1].
* Допустимо работно време за програмата: 0.1 секунди. Допустима памет: 16 MB.

### Примери

|  |  |  |
| --- | --- | --- |
| **Вход** | **Изход** | **Коментари** |
| 1 2 3 4 5  2 \* 2  2 / 2  1 – 2  -2 & 1  stop | [0, 2, 6, 0, 2] | Start at index 0. First command has offset 2, we end up at index 2 (which is the number 3). Multiply it by 2 to get 6. Next command has offset 2; current index is 2, so target index is 2 + 2 = 4 (which is the number 5). Divide by 2 (integer division) to get 2. Next command has offset 1, current index is 4, target index is 5 (1 + 4) which ends up out of bounds, so we start at 0 again, thus target index is 0 which is the number 1. Subtract 2 to get -1, which is not valid, so it becomes 0 instead. Final command has offset -2, target index is -2, which is out of bounds, so start at the end of the array to arrive at index 3 which is the number 4. Apply bitwise AND with 1 which results with 0 at this position. Final array is [0, 2, 6, 0, 2]. |