Кеш памет

На всеки от вас е позната концепцията за кеш памет - пазят се N на брой елементи, които са най-често използвани. Когато правим заявка за търсене, първо се проверява дали елементът, който търсим, е в кеша, и чак ако не го намерим, търсим в паметта.

Реализирайте кеш памет, която да отговаря на следните заявки:

$put < x >< y >$ - добавя елемент с ключ x и стойност y, ако го няма в кеша. Ако го има в кеша, обновява стойността му. И в двата случая това става най-скоро използваният елемент. Ако се опитаме да добавим елемент, когато капацитетът на кеш паметта е запълнен, изхвърляме от нея най-отдавна използвания елемент.

$get < x >$ - отпечатва стойността на елемента с ключ x, ако го има в кеша. Отново става най-скоро използваният елемент. Ако го няма в кеша, отпечатва -1.

Вашата кеш памет за съжаление не е от най-мощните и от време на време прегрява. Това се случва след изпълнението на всяка K-та заявка. Когато прегрее, се изтрива най-отдавна използвания елемент.

Input Format

Първият ред на стандартния вход съдържа три цели числа - капацитета на кеш паметта - N, броя на заявките - Q и броя на заявки, след които прегрява - K. На следващите Q реда получавате заявки от типа $put < x >< y >$ или $get < x >$.

Constraints

$1 \leq N \leq 3000$
$1 \leq Q, K \leq 2 * 10^5$
$0 \leq x \leq 10^4$
$0 \leq y \leq 10^5$

Output Format

Трябва да изведете отговорите на всяка get заявка на нов ред.

Sample Input 0

```
2 9 4
put 1 1
put 2 2
get 1
put 3 3
get 2
put 4 4
get 1
get 3
get 4
```

Sample Output 0

```
1
-1
-1
3
-1
```

Explanation 0

Капацитетът на кеш паметта е 2. Добавяме (1,1) и (2,2) и ключовете, записани в кеша, изглеждат така:

2 1

Търсим елемента с ключ 1 - намираме го и той става последният използван:

1 2

Добавяме (3,3), надхвърляме капацитета и изтриваме последния елемент:

3 1

Но това е 4тата заявка, така че изтриваме и (1,1):

3

Търсим 2 - не го намираме и връщаме -1. Добавяме (4,4):

4 3

Търсим 1, него го няма, 3 го намираме, но това отново е 4та заявка, затова изтриваме (4,4) и не го намираме.