

F. Lại là truy vấn trên cây

time limit per test: 1.75 seconds

memory limit per test: 256 megabytes

input: query.inp

output: query.out

Cho một cây gồm n đỉnh. Các đỉnh được đánh số từ 1 tới n . Đỉnh 1 là gốc của cây. Trên mỗi đỉnh lưu một số nguyên. Ban đầu giá trị ở tất cả các đỉnh là 0. Bạn cần thực hiện q truy vấn thuộc một trong ba dạng sau:

- *add u δ*: Ta thay đổi giá trị các đỉnh thuộc cây con gốc u theo quy tắc:
 - Giá trị của đỉnh u được tăng lên $δ$.
 - Giá trị các đỉnh là con trực tiếp của u bị giảm đi $δ$.
 - Giá trị các đỉnh là con trực tiếp của con trực tiếp của u được tăng lên $δ$.
 - Giá trị các đỉnh là con trực tiếp của con trực tiếp của con trực tiếp của u bị giảm đi $δ$...
- *get u*: Tính giá trị của đỉnh u .
- *pos u k*: Tính giá trị lớn thứ k trong tập giá trị các đỉnh thuộc cây con gốc u .

Yêu cầu: In ra kết quả các truy vấn loại 2 và 3. Với truy vấn loại 3, nếu cây con gốc u có ít hơn k đỉnh, kết quả của truy vấn này là 0.

Input

- Dòng đầu tiên chứa số nguyên t ($1 \leq t \leq 4$) — số thứ tự của subtask chứa test này.
- Dòng thứ hai chứa số nguyên n ($2 \leq n \leq 3 \cdot 10^5$) — số đỉnh của cây.
- Dòng thứ ba chứa $n - 1$ số nguyên p_2, p_3, \dots, p_n ($1 \leq p_i \leq i - 1$), trong đó p_i là cha trực tiếp của đỉnh i .
- Dòng thứ tư chứa số nguyên q ($1 \leq q \leq 3 \cdot 10^5$) — số truy vấn cần thực hiện.
- q dòng cuối cùng, mỗi dòng mô tả một truy vấn theo một trong ba định dạng *add u δ*, *get u* hoặc *pos u k*. Các tham số trên thỏa mãn $1 \leq u \leq n$, $0 \leq |δ| \leq 7000$ và $1 \leq k \leq 15$.

Output

In ra kết quả của các truy vấn loại 2 và loại 3, theo thứ tự chúng xuất hiện trong input. Các số được viết trên một dòng, ngăn cách với nhau bởi dấu cách.

Scoring

- Subtask 1 (29 điểm): $1 \leq n, q \leq 8000$.
- Subtask 2 (17 điểm): Không có truy vấn loại 1 (*add*).
- Subtask 3 (23 điểm): Trong mọi truy vấn loại 3 (*pos*), $k = 1$.
- Subtask 4 (31 điểm): Không có ràng buộc gì thêm.

Examples

input

1

7

1 2 2 4 1 6

18

add 1 1

get 1

get 2

get 3

get 4

get 5

get 6

get 7

add 2 5

add 7 -3

pos 1 1

pos 1 2

pos 1 3

pos 1 4

pos 1 5

pos 1 6

pos 1 7

pos 1 8

output

1 -1 1 1 -1 -1 1 4 4 1 -1 -2 -4 -4 0

input

1

2

1

1

get 1

output

0

input

1

3

1 1

5

add 2 7000

add 3 -7000

pos 1 1

pos 1 2

pos 1 3

output

7000 0 -7000