

Simple Text Editor

Implement a simple text editor. The editor initially contains an empty string, S . Perform Q operations of the following 4 types:

1. $append(W)$ - Append string W to the end of S .
2. $delete(k)$ - Delete the last k characters of S .
3. $print(k)$ - Print the k^{th} character of S .
4. $undo()$ - Undo the last (not previously undone) operation of type 1 or 2, reverting S to the state it was in prior to that operation.

Example

$S = \text{'abcde'}$

$ops = [\text{'1 fg'}, \text{'3 6'}, \text{'2 5'}, \text{'4'}, \text{'3 7'}, \text{'4'}, \text{'3 4'}]$

operation index	S	ops[index]	explanation
-----	-----	-----	-----
0	abcde	1 fg	append fg
1	abcdefg	3 6	print the 6th letter - f
2	abcdefg	2 5	delete the last 5 letters
3	ab	4	undo the last operation, index 2
4	abcdefg	3 7	print the 7th characgter - g
5	abcdefg	4	undo the last operation, index 0
6	abcde	3 4	print the 4th character - d

The results should be printed as:

```
f
g
d
```

Input Format

The first line contains an integer, Q , denoting the number of operations.

Each line i of the Q subsequent lines (where $0 \leq i < Q$) defines an operation to be performed. Each operation starts with a single integer, t (where $t \in \{1, 2, 3, 4\}$), denoting a type of operation as defined in the *Problem Statement* above. If the operation requires an argument, t is followed by its space-separated argument. For example, if $t = 1$ and $W = \text{"abcd"}$, line i will be `1 abcd`.

Constraints

- $1 \leq Q \leq 10^6$
- $1 \leq k \leq |S|$
- The sum of the lengths of all W in the input $\leq 10^6$.

- The sum of k over all delete operations $\leq 2 \cdot 10^6$.
- All input characters are lowercase English letters.
- It is guaranteed that the sequence of operations given as input is possible to perform.

Output Format

Each operation of type **3** must print the k^{th} character on a new line.

Sample Input

```
STDIN      Function
-----
8          Q = 8
1 abc     ops[0] = '1 abc'
3 3       ops[1] = '3 3'
2 3       ...
1 xy
3 2
4
4
3 1
```

Sample Output

```
c
y
a
```

Explanation

Initially, S is empty. The following sequence of 8 operations are described below:

1. $S = ""$. We append *abc* to S , so $S = "abc"$.
2. Print the 3^{rd} character on a new line. Currently, the 3^{rd} character is c.
3. Delete the last 3 characters in S (*abc*), so $S = ""$.
4. Append *xy* to S , so $S = "xy"$.
5. Print the 2^{nd} character on a new line. Currently, the 2^{nd} character is y.
6. Undo the last update to S , making S empty again (i.e., $S = ""$).
7. Undo the next to last update to S (the deletion of the last 3 characters), making $S = "abc"$.
8. Print the 1^{st} character on a new line. Currently, the 1^{st} character is a.