**Uncle Grandpa and heap addition**

After winning the Binary Tree Challenge last week, this week our Grandpa is doing experiments with Heap. He wants to build a special heap that supports the following 3 operations:

* : Insert the number into the heap
* : Increase all elements in the heap by
* : Pop the element with the smallest value out of the heap and print its value

The special heap is empty before the 1st operation is executed.

Actually, our Grandpa has already finished implementing that special heap, but for the purpose of writing the statement, he will still ask for your help. Can you help him?

## Input

The first line contains a single integer – the number of operation

In the next lines, each will contain one query of the above 3 types. In all query,

## Output

For each query of the 3rd type, if the heap is empty, print . Else, print out a single integer is the answer to the query.

## Examples

|  |  |
| --- | --- |
| Input (add1.in) | Output (add1.out) |
| 6  1 5  1 8  2 3  1 7  3  3 | 7  8 |

## Explanation:

We have the state of the heap after each query as follows:

After the 1st query:

After the 2nd query:

After the 3rd query:

After the 4th query:

After the 5th query: (7 is the smallest element hence got popped out) print

After the 6th query: (8 is the smallest element hence got popped out) print

## Note:

1. A skeleton file has been given to help you. You should not create a new file or rename the file provided. You should develop your program using this skeleton file.
2. You are free to define your own helper methods and classes (or remove existing ones) if it is suitable but you must put all the new classes, if any, in the same skeleton file provided

## Skeleton File

You can find the skeleton file Add.java in the lab package.