# CS2233: Data Structures
## Assignment 3
## 23rd September, 2018

### Problem Statement

- Input: A set $S = \{a_1, a_2, \ldots, a_n\}$ of natural numbers given as a sequence. Additionally, several requests to perform operations on $S$.
- Goal: Serve the following requests:
  1. Add an input number $x$ into set $S$. (i.e., $S \leftarrow S \cup \{x\}$)
  2. Find successor of a given number $x$ in $S$.
  3. Delete a given number $x$ from $S$. (i.e., $S \leftarrow S \setminus \{x\}$)
  4. Search for a given number $x$ in $S$.
  5. Print the set.

### Input Format

Each line of the input starts with one of six symbols:

- 'N' (stands for new set)
- '+' (Add element to set)
- '>' (Find successor)
- '–' (Delete element)
- 'S' (search within most recent set)
- 'P' (pre-order traversal)

Format in detail: For lines starting with 'N', 'S', or 'P', refer to format specification in Assignment 2B. For the remaining:

- Lines that start with '+' or '>' or '–' are followed by a space and an $x \in \mathbb{N}$ followed by a '\n' character.
- End of input is indicated by EOF.

### Output Format

- If the input line started with 'N' or 'S' or 'P', the output format is exactly as per specification in Assignment 2B.
- If input line started with '+' or '–', then no corresponding output.
- If input line was "> $x$", then output the smallest number in the set $S$ larger than $x$ even if $x \notin S$. If all numbers in $S$ are smaller than $x$, then output $-1$.

### Implementation rules

In addition to rules specified in Assignment 2B:

- All requests are to be handled with respect to the most recent set built.

### Design decisions

- To keep your program modular, you might want to write a separate procedure for splicing out a node from your BST.
- As discussed in class, you could replace a deleted node by its successor or its predecessor. Make this choice explicit in your code by leaving a comment at the relevant location.

### Other Remarks

- You are encouraged to use and modify your own code from previous assignments.
- **Deadline:** Midnight of 30th September 2018.

### Example

Input:

```
---------------------------
N 12 15 9 34 8 14
+ 10
+ 78
+ 3
S 16
S 14
> 12
> 11
P
- 14
> 12
- 34
S 14
S 78
> 78
+ 68
+ 80
> 15
S 68
P
- 12
P
N 90
+ 23
+ 26
+ 108
> 23
S 90
P
-90
P

---------------------------
```

Output:

```
---------------------------
-1
10
14
12
12 9 8 3 10 15 14 34 78
15
-1
11
-1
68
110
12 9 8 3 10 15 78 68 80
15 9 8 3 10 78 68 80
26

90 23 26 108
108 23 26

---------------------------
```