

Byteburg Inventory System

In Byteburg, the Inventory System has k bins to store various items. Each item is assigned to a bin using the formula $h(x) = x \bmod k$. When two items are meant for the same bin, the Inventory Clerk uses Quadratic Probing to find the next available bin.

As items are added, if the system reaches over 75% capacity, the Inventory Clerk will perform an upgrade. This upgrade involves rehashing the items.

Rehashing is the process of reorganizing the hash table when the number of items exceeds a certain capacity threshold. In this system, rehashing occurs when more than 75% of the bins are occupied. The Inventory Clerk doubles the number of bins (from k to $2k$) and reinserts all the existing items(input order) into the newly expanded hash table according to the new bin size. This process reduces the likelihood of collisions and the need for quadratic probing.

Tasks

- **Inserting Items into the Hash Table** (`HashInsert(T, val)`): Each time you insert an item 'val' into the hash table T , the table might rearrange itself automatically based on the hash function and collision resolution method.
- **Displaying the Hash Table** (`DisplayTable(T)`): This function will output the current state of the hash table T , showing the current elements stored in it.

Input Format

- The first line contains an integer k , representing the initial number of bins in the hash table.
- Each subsequent line contains a character from { 'i', 'd', 'e' } followed by zero or one integer value, where the integer value $\in [1, 10^9]$.
- The character 'i' is followed by a positive integer value 'val' separated by a space. Perform `HashInsert(T, val)` operation.
- The character 'd' displays the current state of the hash table T .
- The character 'e' terminates the sequence of operations.

Output Format

- The output of each command (if any) should be printed on a separate line. However, no output is printed for the 'i' and 'e' commands.
- For the 'd' command, display the current state of the hash table T , showing the contents of each bin. The resulting table should be displayed with bin values separated by spaces.
- If a bin is empty, it should be represented by -1.

Sample Test Cases

Input 1:

```
8
i 15
i 23
i 32
i 9
i 14
i 21
d
i 30
i 18
d
e
```

Output 1:

```
23 32 9 -1 -1 21 14 15
32 -1 30 18 -1 21 -1 23 -1 9 -1 -1 -1 -1 14 15
```

Input 2:

```
5
i 10
i 12
i 23
d
i 20
i 27
d
e
```

Output 2:

```
10 -1 12 23 -1
10 20 12 23 -1 -1 -1 27 -1 -1
```