

# Hash Table

## Problem ID: hashtable2

Write your own implementation of a hash table, specialized to store 32-bit signed integers as both keys and values in an associative manner.

The data structure should support the following operations:

- Default construction - initializes an empty hash table. This is not tested explicitly here!
- Assignment and copy construction - must properly copy the contents of another instance of the data structure. Ensure the two instances do not share memory afterwards!
- Insert - must insert a key-value pair to the hash table. If the key is present, then nothing is inserted.
- Erase - must remove the given key from the hash table.
- Element access - must provide access to the value associated with a given key.
- Size - must provide the size of the hash table.

You must avoid any memory leaks or other memory errors in your implementation.

### Input

The input starts with a line containing an integer  $q$ , representing the number of lines that follow. Each line will represent an operation that is run. The lines have the following format: `<id> <operation> [arg1] [arg2]`

The instance ID will be an integer from 1 to 1,000, representing an instance of the data structure. Each instance should be default initialized at the start as an empty hash table.

The operations are the following:

- `a` - construct a copy of the hash table, takes integer argument for the instance ID of the hash table to copy
- `i` - insert a key-value pair to the hash table if the key does not exist, otherwise do nothing
- `e` - erase a given key from the hash table which is guaranteed to exist
- `g` - element access, get, output the value associated with the given key if it exists, see output section
- `s` - element access, set, set the value associated with a given key which is guaranteed to exist
- `z` - output size of hash table, see output section

You may assume requested operations will not cause an error in a correctly implemented data structure. For example, a pop operation on an empty hash table will not occur in the input.

### Output

For each get operation, output a line with the associated value if the key exists, otherwise output a line with a single dash.

For each size operation, output a line with the size of the hash table.

**Sample Input 1**

```
11
1 i 2 0
1 z
1 g 3
1 i 3 -4
1 g 3
1 s 3 -3
1 i 3 7
1 g 3
1 e 3
1 g 3
1 z
```

**Sample Output 1**

```
1
-
-4
-3
-
1
```

**Sample Input 2**

```
10
1 i 0 1
1 i 2 4
2 i -1 1
2 i 2 -2
1 a 1
3 a 1
1 a 2
1 g 2
2 g 2
3 g 2
```

**Sample Output 2**

```
-2
-2
4
```