2. JavaScript: Staff List

The task is to create a class StaffList. The class will manage a collection of staff members, where each member is uniquely identified by a name. The class must have the following methods:

1. add(name, age):

- Parameters string name and integer age are passed to this function.
- o If age is greater than 20, it adds the member with the given name to the collection.
- Else if age is less than or equal to 20, it throws an Error with the message 'Staff member age must be greater than 20'.
- o It is guaranteed that at any time, if a member is in the collection, then no other member with the same name will be added to the collection.

2. remove(name):

- o If the member with the given name is in the collection, it removes the member from the collection and returns true.
- Else if the member with the given name is not in the collection, it does nothing and returns false.

3. getSize():

returns the number of members in the collection.

Your implementation of the class will be tested by a stubbed code on several input files. Each input file contains parameters for the functions calls. The functions will be called with those parameters, and the result of their executions will be printed to the standard output by the provided code. The stubbed code prints values returned by the remove(name) and getSize() functions, and it also prints messages of all the cached errors.

▼ Input Format For Custom Testing

The first line contains an integer, n, denoting the number of operations to be performed.

Each line i of the n subsequent lines (where $0 \le i < n$) contains space-separated strings, such that the first of them is the function name and the remaining ones, if any, are parameters for that function.

▼ Sample Case 0

Sample Input For Custom Testing

```
add John 25
add Robin 23
getSize
remove Robin
getSize
```

Sample Output

```
2
true
1
```

Explanation

There are 2 staff members, 'John' and 'Robin', who are added by calling the *add* function twice. *getSize* is then called and returns the number of members in the collection, which is 2. Then, the staff member 'Robin' is removed from the list by calling the *remove* function, and since the given name is in the collection, the return value true is printed. Finally, *getSize* is called, which prints the size of the collect, which is now 1 because 'Robin' was removed.

▼ Sample Case 1

Sample Input For Custom Testing

```
5
add John 20
add Robin 10
getSize
remove Robin
getSize
```

Sample Output

```
Error: Staff member age must be greater than 20
Error: Staff member age must be greater than 20
0
false
```

Explanation

The function *add* is called to add 'John' and 'Robin', but in both cases, an error is thrown since *age* <= 20. Then, the *getSize* function is called and returns 0, which is the size of the collection, which is then printed. Next, the *remove* function is called to remove 'Robin', but since this staff member does not exist, false is returned and printed. Finally, the *getSize* function is called, which again returns 0, which is the size of the collection.