National Taiwan University, Department of Electronic Engineering

Data Structure, Spring 2020 Assignment #5

1 Problem Statement

In this homework, you are going to implement a hash table. A common problem when implementing hash functions is collision, and one of the ways to solve collisions is to chain the data up. Now, we give you a hash function $f(key) = key \ (mod\ 1001)$. Please implement a hash table with 1001 slots, and rules as below.

At first, the hash table is empty. However, there a some commands as below:

1. **Insert n**: Please insert the item n into the hash table. If a collision occurs, please put the latest inserted item into the front of the linked list.
2. **Look k**: Please print out all the items of the kth slot (there should be space between numbers, and there should be next-line in the end). If there are no items, please output Null.
3. **Delete n**: If n is in the hash table, delete n (if there are more than one element, delete the latest one). If n is not in the hash table, please output Error.
4. **Search n**: If n is in the hash table, output Yes. If not, output No.
5. **End**: end of input

Note that the keys in our test cases are strings. When calculating the hash, please convert it into int with 27-carry. For example, **abc** should be converted into $27^2 * 1 + 27^1 * 2 + 27^0 * 3$. Also, all letters are seen as lower case.

2 Input/Output

Below is a sample of input and output:

Sample Input :

```
Insert abc
Insert ABc
Look 786
Look 787
Insert AbC
Insert aBC
Look 786
Search abc
Delete abc
Look 786
Search abc
Delete ABC
Look 786
Search abc
Delete ABc
Insert ABd
Look 787
End
```

Sample Output :

```
ABc abc
Null
aBC AbC ABc abc
Yes
aBC AbC ABc
No
Error
aBC AbC ABc
No
ABd
```

3 Evaluation

We have provided a code file **hash.py**. We have done the main function for you, and please finish the files TODO part. You can test your code by running **python3 hash.py –input ./input –output ./output**

Do not modify the interface of the functions, but you can add your own functions.

4 Submission

Please put your codes (including main.py or any other code files) into a directory named **studentID** and compress the directory into studentID.zip and upload studentID.zip to ceiba. The homework is due on **6/15**, at **4:00 am**.