Muhindo Mussa Justin 18/December/2022

http://www.spoj.com/problems/HASHIT/

# HASHIT - Hash it!

#hash-table #hashing

Your task is to calculate the result of the hashing process in a table of 101 elements, containing keys that are strings of length at most 15 letters (ASCII codes '*A*',...,'*z*'). Implement the following operations:

- find the index of the element defined by the key (ignore, if no such element),
- insert a new key into the table (ignore insertion of the key that already exists),
- delete a key from the table (without moving the others), by marking the position in table as *empty* (ignore non-existing keys in the table)

When performing find, insert and delete operations define the following function:
*integer Hash(string key)*,
which for a string $key=a_1...a_n$ returns the value:
$Hash(key)=h(key)$ mod 101, where
$h(key)=19*(ASCII(a_1)*1+...+ASCII(a_n)*n)$.
Resolve collisions using the open addressing method, i.e. try to insert the key into the table at the first free position: $(Hash(key)+j^2+23*j)$ mod 101, for $j=1,...,19$. After examining of at least 20 table entries, we assume that the insert operation cannot be performed.

## Input

*t* [the number of test cases <= 100]
$n_1$ [the number of operations (one per line)[<= 1000]
ADD:string
[or]
DEL:string [other test cases, without empty lines betwee series]

## Output

For every test case you have to create a new table, insert or delete keys, and write to the output:
the number of keys in the table [first line]
index:key [sorted by indices]

## Example

```
Input:

1

11
```

```
ADD:marsz

ADD:marsz

ADD:Dabrowski

ADD:z

ADD:ziemii

ADD:wloskiej

ADD:do

ADD:Polski

DEL:od

DEL:do

DEL:wloskiej



Output:

5

34:Dabrowski

46:Polski

63:marsz

76:ziemii

96:z
```

```
using System;
using System.Text;
namespace HASHIT
{
	class MainClass
```

```
{
	public static void Main(string[] args)
	{
		int m = Convert.ToInt32(System.Console.ReadLine());
		for (int i = 0; i < m; i++)
		{
			string[] table = new string[101];
			int n = Convert.ToInt32(System.Console.ReadLine());
			for (int ii = 0; ii < n; ii++)
			{
				string my_line = System.Console.ReadLine();
				string oper = my_line.Substring(0, 3);
				string key = my_line.Substring(4);
				int mx= FindKey(table, key);

				if (oper == "ADD")
				{
					if (mx == -1)
					{
					mx = FindNextOpenAddress(table, Hash(key));
						if (mx >= 0)
						{
							table[mx] = key;
						}
					}
				}
				else
				{
					if (mx >= 0)
					{
						table[mx] = string.Empty;
					}
				}
			}
			StringBuilder sb = new StringBuilder();
			int count = 0;
			for (int j = 0; j < 101; j++)
			{
				if (!string.IsNullOrEmpty(table[j]))
				{
					count++;
					sb.AppendLine(j + ":" + table[j]);
				}
			}
			Console.WriteLine(count);
			Console.Write(sb.ToString());
		}
	}
	public static int Hash(string key)
	{
		int ret = 0;
		ret = h(key) % 101;
		return ret;
	}
	public static int h(string key)
	{
		int ret = 0;
		int cnt = key.Length;
```

```
                for (int i = 0; i < cnt; i++)
                {
                        ret += (int)key[i] * (i + 1);
                }
                return ret * 19;
        }
        public static int FindKey(string[] table, string key)
        {
                int ix = Hash(key);
                if (table[ix] == key)
                        return ix;
                for (int j = 1; j < 20; j++)
                {
                        int newix = (ix + j * j + 23 * j) % 101;
                        if (table[newix] == key)
                        {
                                return newix;
                        }
                }
                return -1;
        }
        public static int FindNextOpenAddress(string[] table, int ix)
        {
                if (string.IsNullOrEmpty(table[ix]))
                        return ix;
                for (int j = 1; j < 20; j++)
                {
                        int newix = (ix + j * j + 23 * j) % 101;
                        if (string.IsNullOrEmpty(table[newix]))
                        {
                                return newix;
                        }
                }
                return -1;
        }
    }
}
```

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| 30589437 | 2022-12-18 11:54:18 | MMJ | Hash it! | **accepted** edit ideone it | 0.07 | 27M | NCSHARP |