**HASH**

**Note: the hash table should be bigger than 30 ,at the least .**

Implement a hash table. The hash function to be used is the “modulo operation”. Resolve collisions by using

a) Chaining.

b) Open Addressing

i. Linear Probing

ii) Double Hashing (Use a convenient function for the secondary hash function)

Your program must support the following functions:

**insert(h, key)** – insert the data specified by key into the hash table specified by h.

 **search(h, key)** – search for the data specified by key in the hash table specified by h.

**Input - Output Format**

The first line contains a single positive integer c, the capacity of the hash table. All modulo operations have to be performed using c.

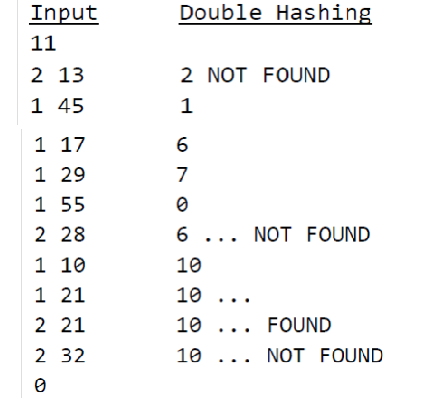
The rest of the input consists of multiple lines, each one containing either one or two integers.

The first integer in the line can be 0, 1, 2, 3 or 4, and each one has its own meaning:

* The integer 0 means stop the program.
* The integer 1 means insert the next integer from the input into the hash table. Output the index at which the data is stored. If open addressing is used, in case of a collision, output the probe sequence (here, the index at which the data will get stored must be printed only once, and at the end of the sequence).
* The integer 2 means search for the next integer from the input into the hash table. Output “FOUND”, if the search is successful. Otherwise, output “NOT FOUND”. If open addressing is used, output the probe sequence, before the message.

**Sample Input and Output**

****

****