All Contests > CSE-g-Week-16 > Implement double hashing

Implement double hashing

Problem Submissions Leaderboard

Write a C program to implement double hashing. The 2 hash functions are defined as follows:

h1(k) = (2k+3) mod m

h2(k) = (3k+1) mod m

Print the elements of the hash table. If no element has been inserted at that particular index then print -1 else print the element inserted.

Note: It is not mandatory for all elements to be successfully inserted into the hash table. If repeated collisions occur and the probing cycle leads back to an already-visited index (indicating that all possible positions have been checked without finding an empty slot), the insertion process for that element should be stopped.

Input Format

m (Hash table size)

n (Number of elements)

Element 1

Element 2

.

.

.

Element n

Constraints

Value of elements >=0

Output Format

Element in index 0

Element in index 1

.

.

.

Element in index (n-1)

Sample Input 0

```
10
8
3
2
9
```

```
6
11
13
7
12
```

Sample Output 0

```
-1
9
-1
11
12
6
-1
2
-1
3
```

Contest ends in 9 hours

Submissions: 1
Max Score: 10
Difficulty: Easy

Rate This Challenge:
☆☆☆☆☆

More

C

```c
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
#define EMPTY -1

int h1(int k, int m) {
    return (2 * k + 3) % m;
}

int h2(int k, int m) {
    return (3 * k + 1) % m;
}

void insert(int hash_table[], int m, int key) {
    int index = h1(key, m);
    int step = h2(key, m);

    int i = 0; // counter
    while (hash_table[index] != EMPTY && i < m) {
        index = (index + step) % m;
        i++;
    }

    if (i < m) {
        hash_table[index] = key; // insert key
    }
}

void printHashTable(int hash_table[], int m) {
    for (int i = 0; i < m; i++) {
        printf("%d\n", hash_table[i]);
    }
}

int main() {
    int m, n;
    scanf("%d", &m);
    scanf("%d", &n);

    int hash_table[m];
    for (int i = 0; i < m; i++) {
        hash_table[i] = EMPTY; // initialize hashtable
    }

    for (int i = 0; i < n; i++) {
        int key;
        scanf("%d", &key);
        insert(hash_table, m, key);
    }

    printHashTable(hash_table, m);

    return 0;
}
```

Line: 43 Col: 36

Upload Code as File □ Test against custom input Run Code Submit Code

Testcase 0 ✓

Congratulations, you passed the sample test case.

Click the **Submit Code** button to run your code against all the test cases.

Input (stdin)

```
10
8
3
2
9
6
11
13
7
12
```

Your Output (stdout)

```
-1
9
-1
11
12
6
-1
2
-1
3
```

Expected Output

```
-1
9
-1
11
12
6
-1
2
-1
3
```