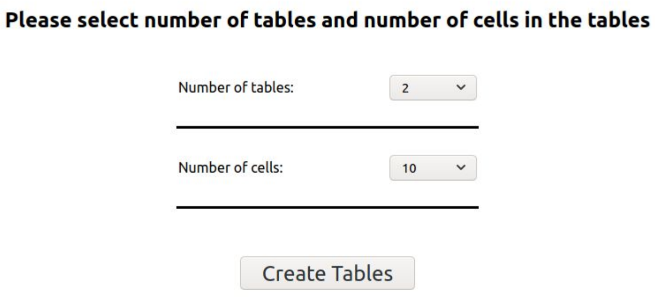
Advanced Data Structure Project 2

Mehmet MUM 150114051

I have used react for this project. HTML/CSS for frontend and JavaScript for hash operations.

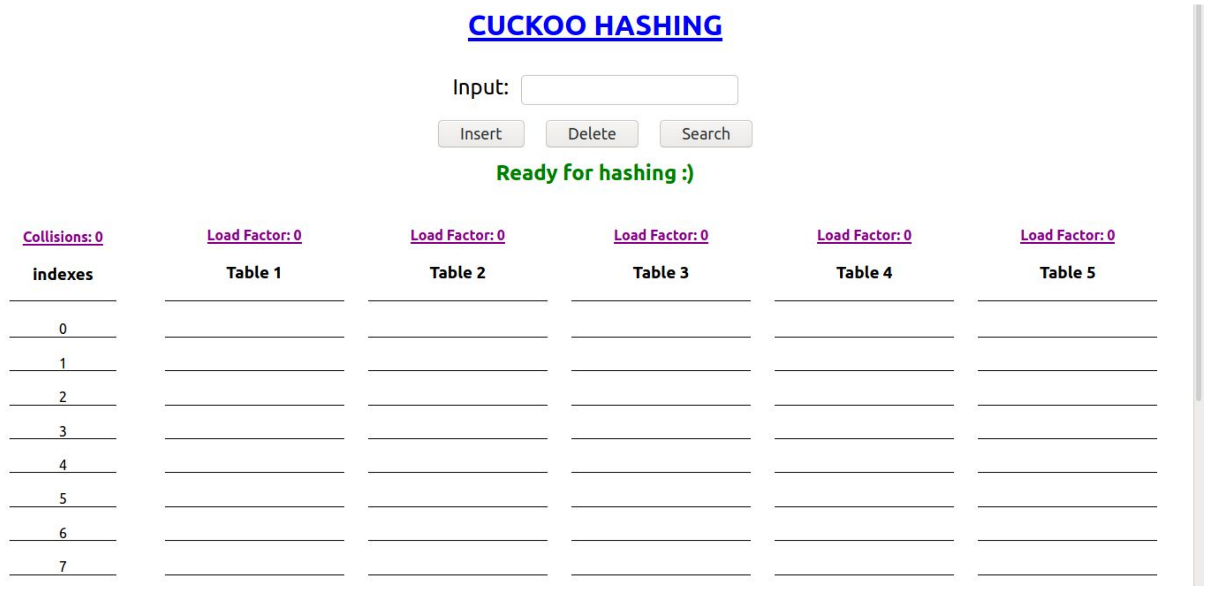
Number of tables and number of cells are taken from user.



Please select number of tables and number of cells in the tables

Number of tables:	2
Number of cells:	10

And then input screen will be created.



CUCKOO HASHING

Input:

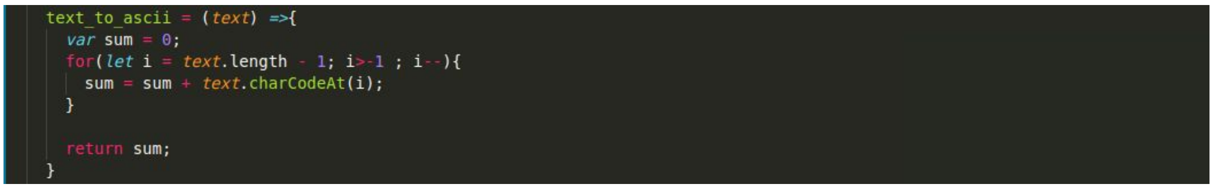
Ready for hashing :)

Collisions: 0	Load Factor: 0	Load Factor: 0	Load Factor: 0	Load Factor: 0	Load Factor: 0
indexes	Table 1	Table 2	Table 3	Table 4	Table 5
0					
1					
2					
3					
4					
5					
6					
7					

Load factor of each table is writing above each tables. User can insert, delete and search. After each insertion, number of collisions will be shown on the indexes.

1 - Input format

User can type anything with max length 20. Program converts input to a key by summing each character's ascii code.



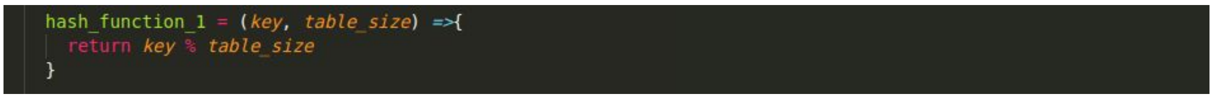
```
text_to_ascii = (text) =>{
  var sum = 0;
  for(let i = text.length - 1; i>-1 ; i--){
    sum = sum + text.charCodeAt(i);
  }

  return sum;
}
```

2 - Hash functions

I created a hash function for each table.

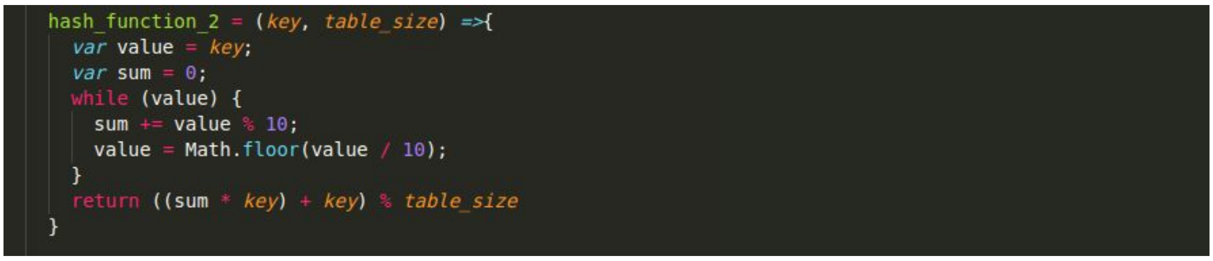
- **TABLE 1**



```
hash_function_1 = (key, table_size) =>{
  return key % table_size
}
```

Just return key % table_size.

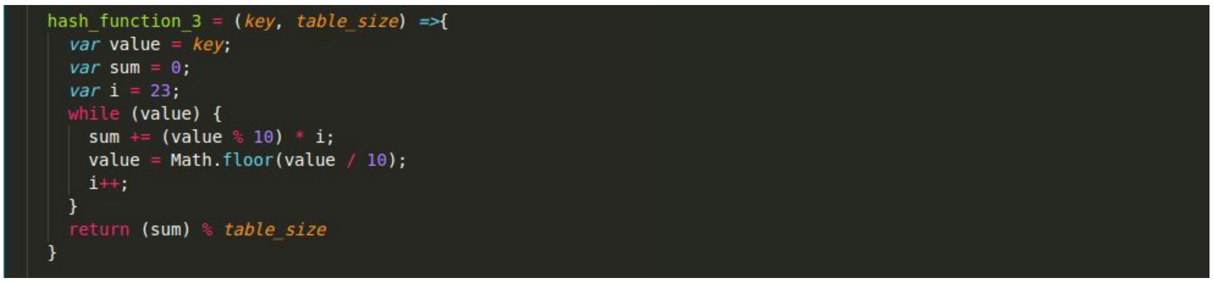
- **TABLE 2**



```
hash_function_2 = (key, table_size) =>{
  var value = key;
  var sum = 0;
  while (value) {
    sum += value % 10;
    value = Math.floor(value / 10);
  }
  return ((sum * key) + key) % table_size
}
```

Sum each digit of key then multiply it with key and then sum the result with key. After all, return result % table_size.

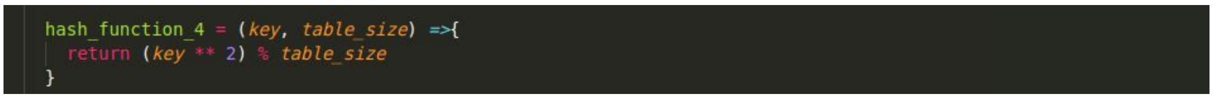
- **TABLE 3**



```
hash_function_3 = (key, table_size) =>{
  var value = key;
  var sum = 0;
  var i = 23;
  while (value) {
    sum += (value % 10) * i;
    value = Math.floor(value / 10);
    i++;
  }
  return (sum) % table_size
}
```

Multiply each digit with a coefficient (starting from 23) then sum them. For instance, key is 103 sum will be 1*23 + 0*24 + 3*25. Then return sum % table_size.

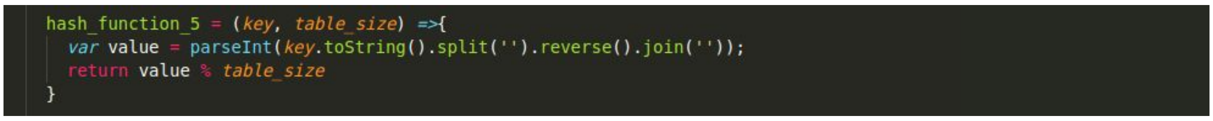
- TABLE 4



```
hash_function_4 = (key, table_size) =>{
  return (key ** 2) % table_size
}
```

return (key*key) % table_size

- TABLE 5

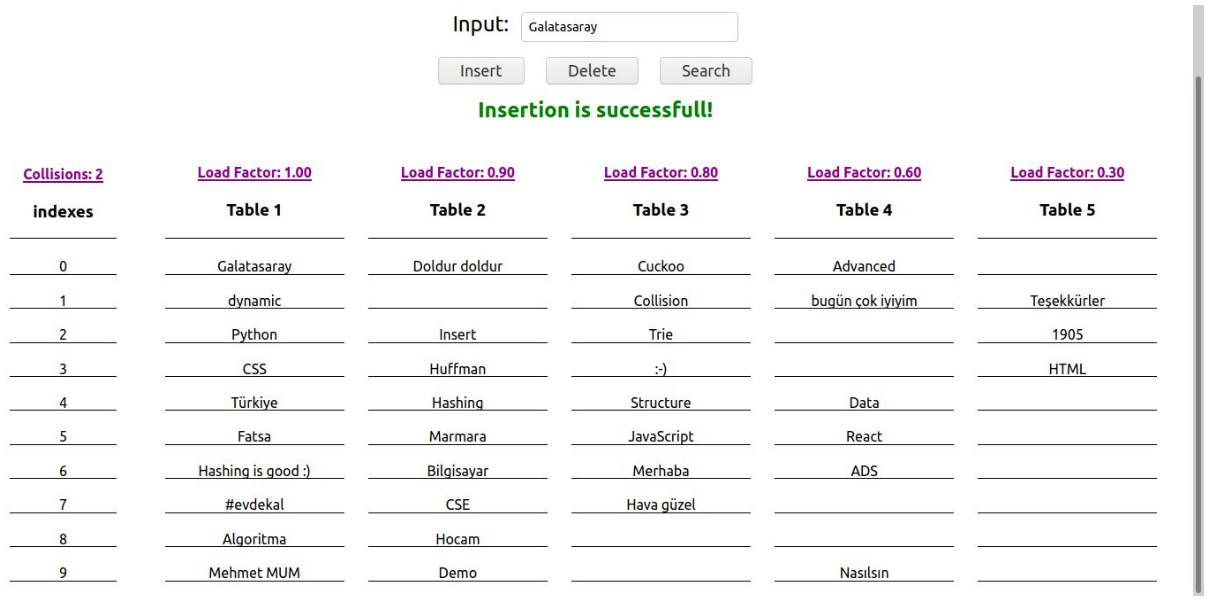


```
hash_function_5 = (key, table_size) =>{
  var value = parseInt(key.toString().split('').reverse().join(''));
  return value % table_size
}
```

get reverse of key and return % table_size

3 - Insertion

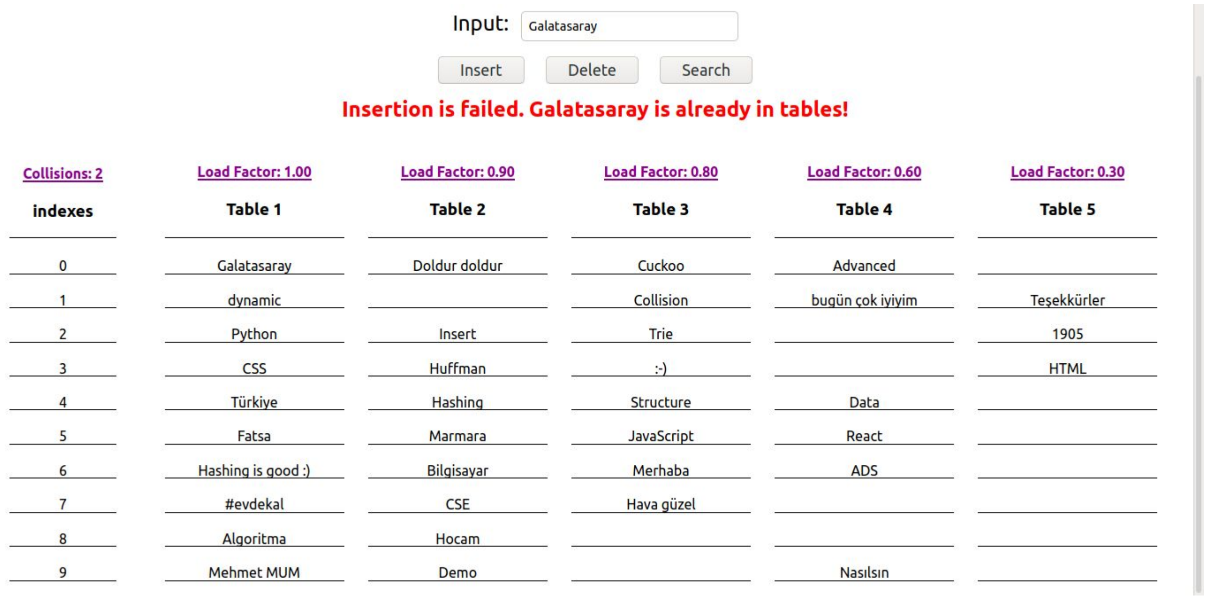
Program inserts an input to cuckoo hashing with starting table 1, if cell is full take the value and insert to next table untill a cell in a table is empty.



Insertion is successfull!

Collisions: 2	Load Factor: 1.00	Load Factor: 0.90	Load Factor: 0.80	Load Factor: 0.60	Load Factor: 0.30
indexes	Table 1	Table 2	Table 3	Table 4	Table 5
0	Galatasaray	Doldur doldur	Cuckoo	Advanced	
1	dynamic		Collision	bugün çok iyiyim	Teşekkürler
2	Python	Insert	Trie		1905
3	CSS	Huffman	:-)		HTML
4	Türkiye	Hashing	Structure	Data	
5	Fatsa	Marmara	JavaScript	React	
6	Hashing is good :)	Bilgisayar	Merhaba	ADS	
7	#evdekal	CSE	Hava güzel		
8	Algoritma	Hocam			
9	Mehmet MUM	Demo		Nasılsın	

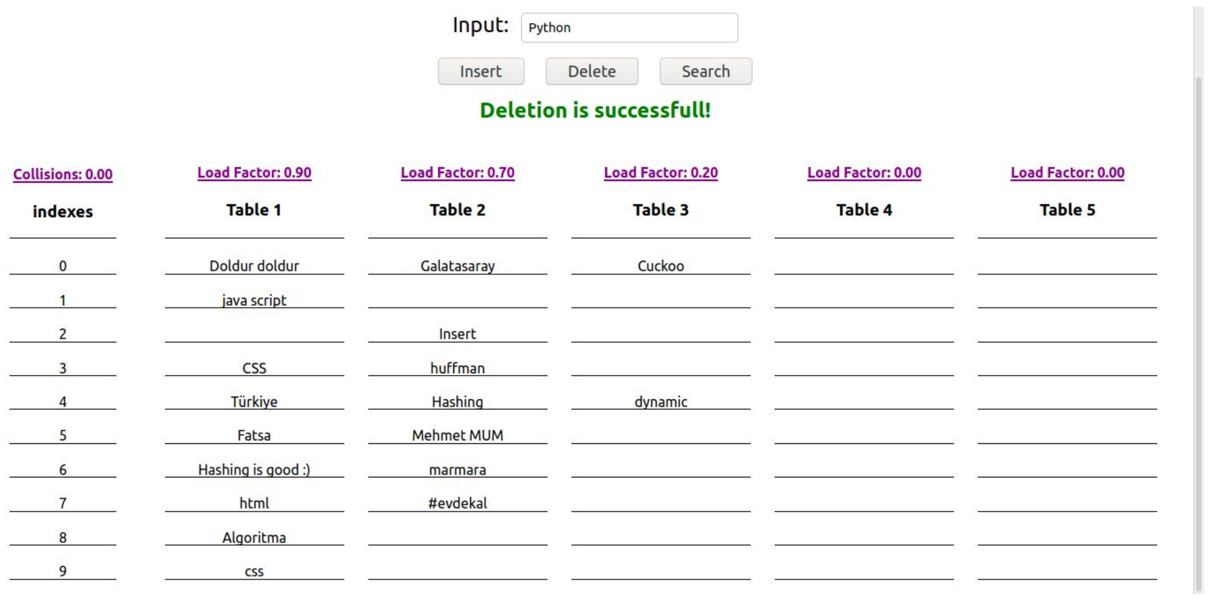
We can insert an input only once.



Insertion is failed. Galatasaray is already in tables!

Collisions: 2	Load Factor: 1.00	Load Factor: 0.90	Load Factor: 0.80	Load Factor: 0.60	Load Factor: 0.30
indexes	Table 1	Table 2	Table 3	Table 4	Table 5
0	Galatasaray	Doldur doldur	Cuckoo	Advanced	
1	dynamic		Collision	bugün çok iyiyim	Teşekkürler
2	Python	Insert	Trie		1905
3	CSS	Huffman	:-)		HTML
4	Türkiye	Hashing	Structure	Data	
5	Fatsa	Marmara	JavaScript	React	
6	Hashing is good :)	Bilgisayar	Merhaba	ADS	
7	#evdekal	CSE	Hava güzel		
8	Algoritma	Hocam			
9	Mehmet MUM	Demo		Nasılsın	

4 - Deletion



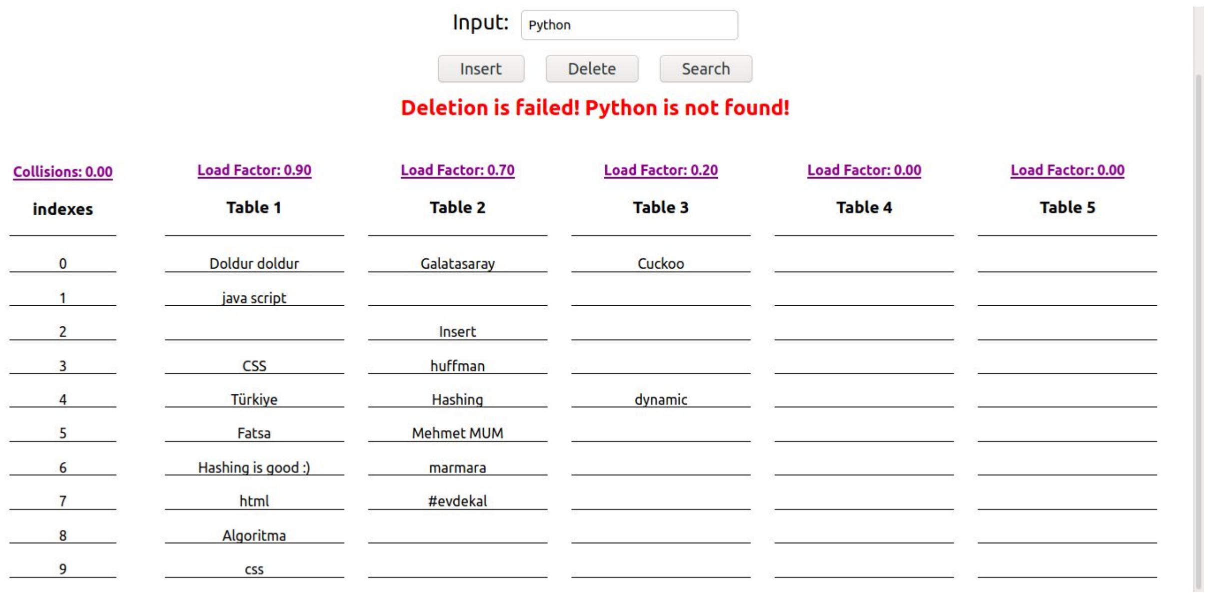
Input: Python

Insert | Delete | Search

Deletion is successfull!

Collisions: 0.00	Load Factor: 0.90	Load Factor: 0.70	Load Factor: 0.20	Load Factor: 0.00	Load Factor: 0.00
indexes	**Table 1**	**Table 2**	**Table 3**	**Table 4**	**Table 5**
0	Doldur doldur	Galatasaray	Cuckoo		
1	java script				
2		Insert			
3	CSS	huffman			
4	Türkiye	Hashing	dynamic		
5	Fatsa	Mehmet MUM			
6	Hashing is good :)	marmara			
7	html	#evdekal			
8	Algoritma				
9	css				

We search for given input on each table if we find it just delete from the cell. For instance Python is deleted from Table 1 (2. index).



Input: Python

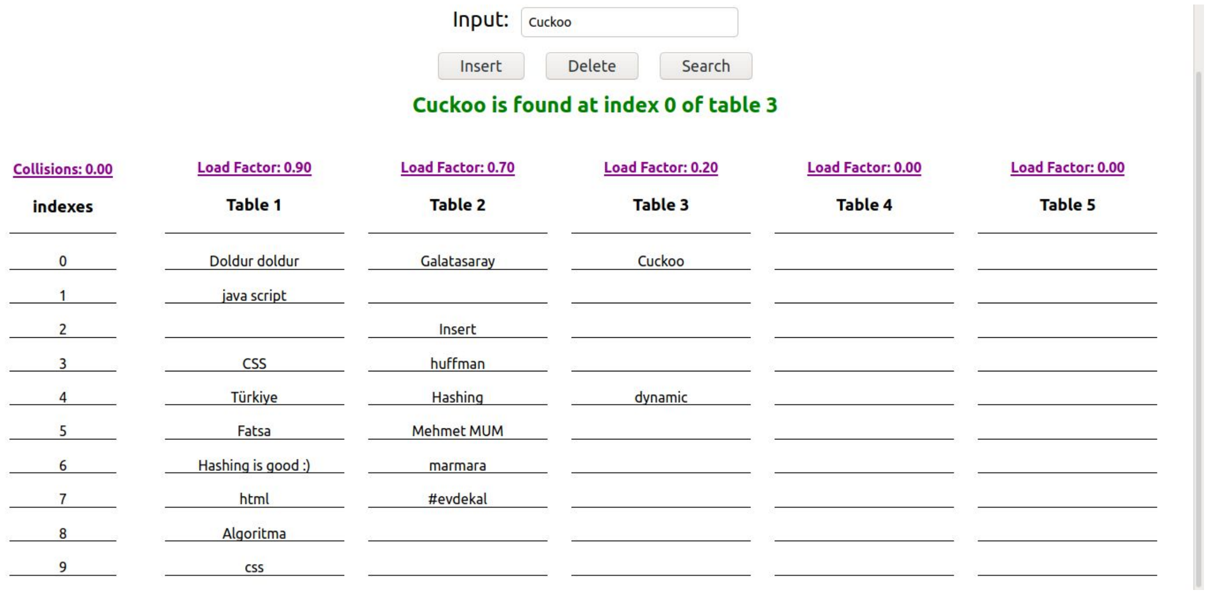
Insert | Delete | Search

Deletion is failed! Python is not found!

Collisions: 0.00	Load Factor: 0.90	Load Factor: 0.70	Load Factor: 0.20	Load Factor: 0.00	Load Factor: 0.00
indexes	**Table 1**	**Table 2**	**Table 3**	**Table 4**	**Table 5**
0	Doldur doldur	Galatasaray	Cuckoo		
1	java script				
2		Insert			
3	CSS	huffman			
4	Türkiye	Hashing	dynamic		
5	Fatsa	Mehmet MUM			
6	Hashing is good :)	marmara			
7	html	#evdekal			
8	Algoritma				
9	css				

We can not delete if given input is not in the tables.

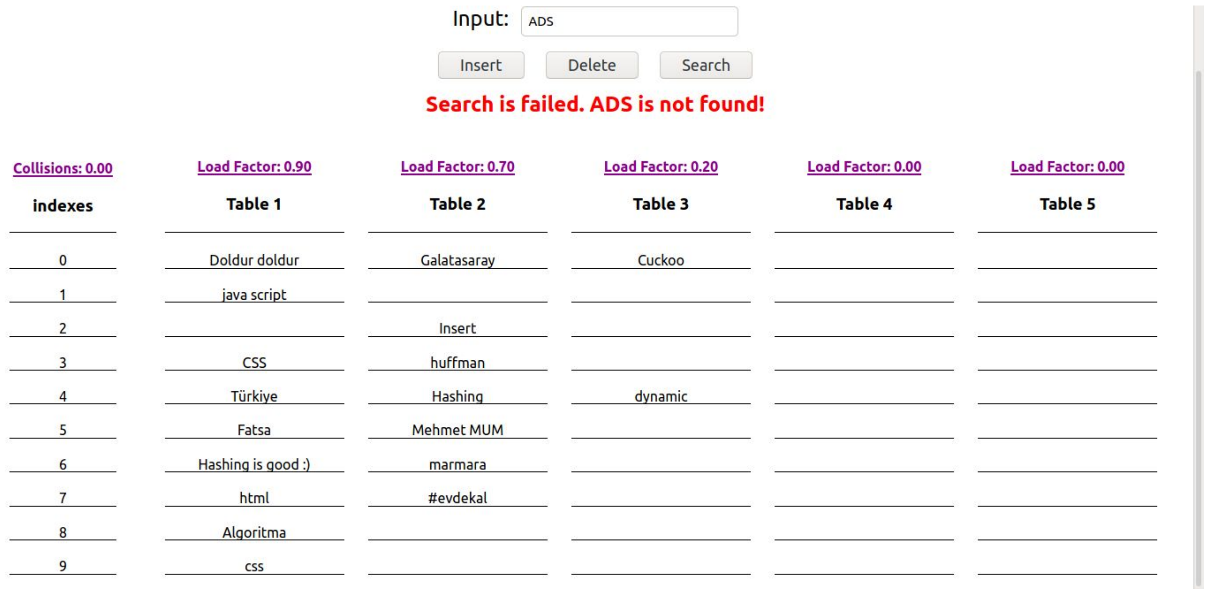
5 - Search



Cuckoo is found at index 0 of table 3

Collisions: 0.00	Load Factor: 0.90	Load Factor: 0.70	Load Factor: 0.20	Load Factor: 0.00	Load Factor: 0.00
indexes	Table 1	Table 2	Table 3	Table 4	Table 5
0	Doldur doldur	Galatasaray	Cuckoo		
1	java script				
2		Insert			
3	CSS	huffman			
4	Türkiye	Hashing	dynamic		
5	Fatsa	Mehmet MUM			
6	Hashing is good :)	marmara			
7	html	#evdekal			
8	Algoritma				
9	css				

Search for Cuckoo and report result.



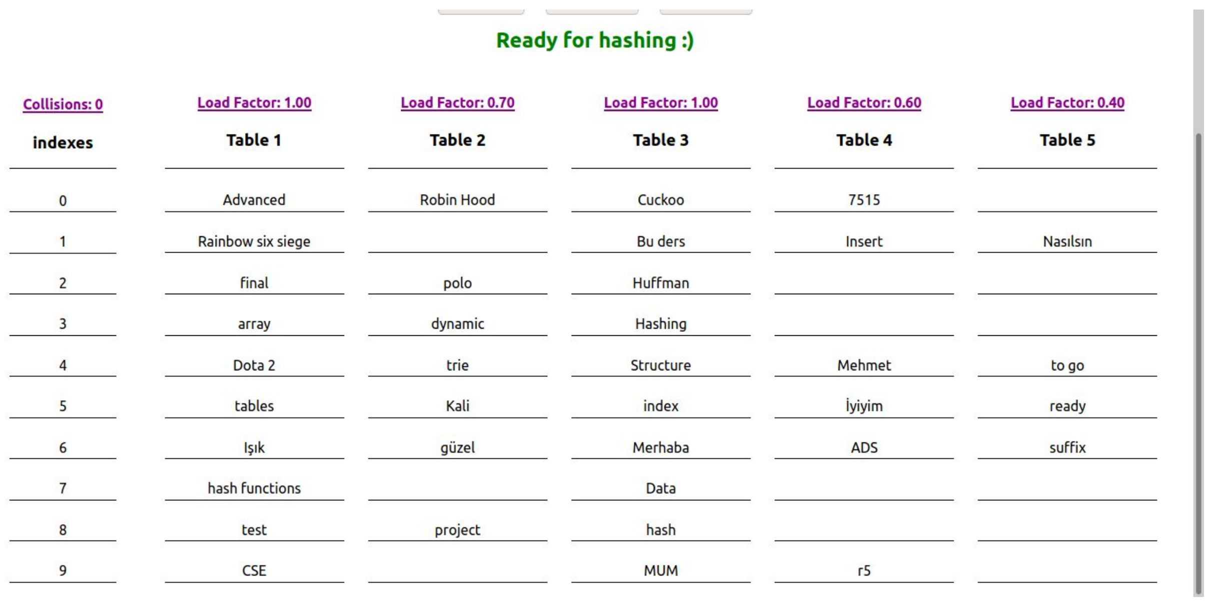
Search is failed. ADS is not found!

Collisions: 0.00	Load Factor: 0.90	Load Factor: 0.70	Load Factor: 0.20	Load Factor: 0.00	Load Factor: 0.00
indexes	Table 1	Table 2	Table 3	Table 4	Table 5
0	Doldur doldur	Galatasaray	Cuckoo		
1	java script				
2		Insert			
3	CSS	huffman			
4	Türkiye	Hashing	dynamic		
5	Fatsa	Mehmet MUM			
6	Hashing is good :)	marmara			
7	html	#evdekal			
8	Algoritma				
9	css				

ADS is not found in the tables

6 - Log

In order to show you what is going on in tables while insertion, i listed operations in tables.



Ready for hashing :)

Collisions: 0	Load Factor: 1.00	Load Factor: 0.70	Load Factor: 1.00	Load Factor: 0.60	Load Factor: 0.40
indexes	**Table 1**	**Table 2**	**Table 3**	**Table 4**	**Table 5**
0	Advanced	Robin Hood	Cuckoo	7515	
1	Rainbow six siege		Bu ders	Insert	Nasılsın
2	final	polo	Huffman		
3	array	dynamic	Hashing		
4	Dota 2	trie	Structure	Mehmet	to go
5	tables	Kali	index	İyiyim	ready
6	Işık	güzel	Merhaba	ADS	suffix
7	hash functions		Data		
8	test	project	hash		
9	CSE		MUM	r5	

This the table before hashing. I will insert “ React “ input to cuckoo hashing

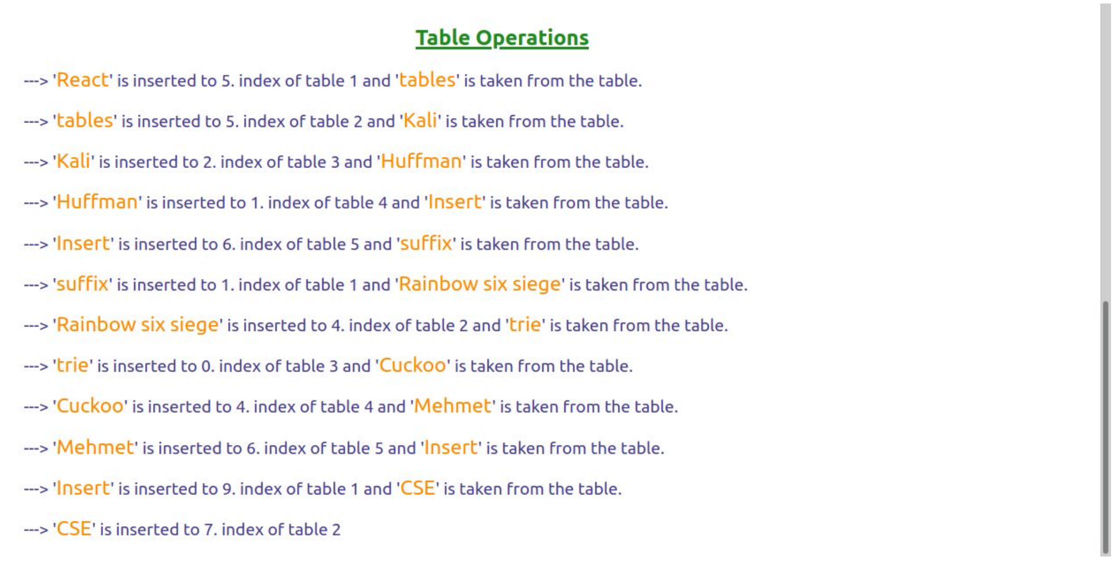
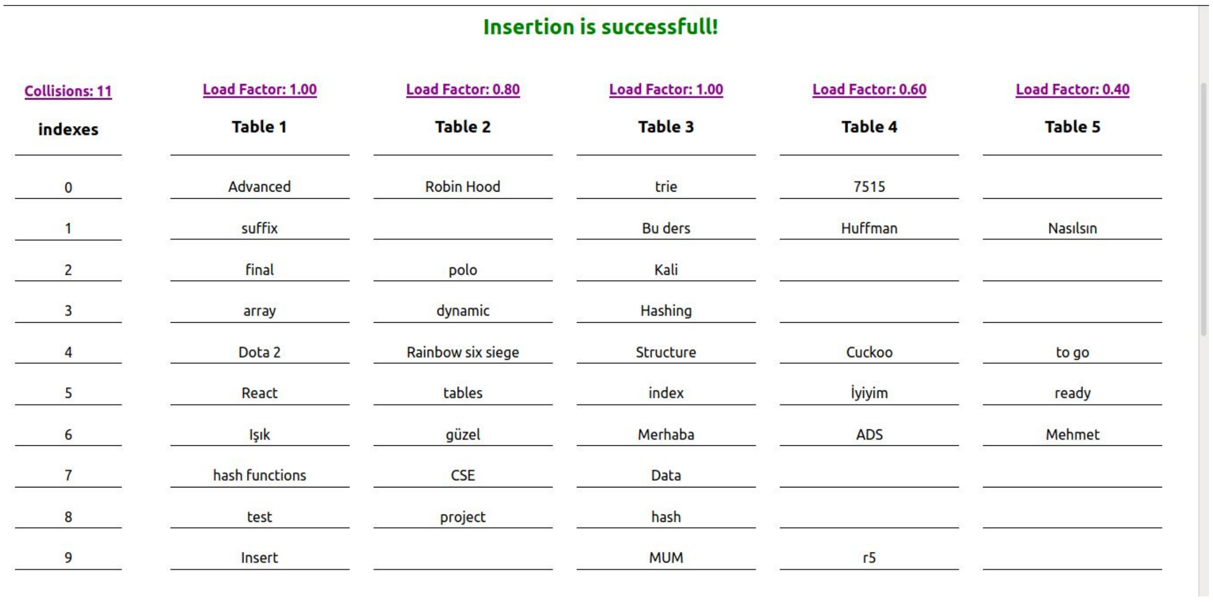


Table Operations

```
---> 'React' is inserted to 5. index of table 1 and 'tables' is taken from the table.
---> 'tables' is inserted to 5. index of table 2 and 'Kali' is taken from the table.
---> 'Kali' is inserted to 2. index of table 3 and 'Huffman' is taken from the table.
---> 'Huffman' is inserted to 1. index of table 4 and 'Insert' is taken from the table.
---> 'Insert' is inserted to 6. index of table 5 and 'suffix' is taken from the table.
---> 'suffix' is inserted to 1. index of table 1 and 'Rainbow six siege' is taken from the table.
---> 'Rainbow six siege' is inserted to 4. index of table 2 and 'trie' is taken from the table.
---> 'trie' is inserted to 0. index of table 3 and 'Cuckoo' is taken from the table.
---> 'Cuckoo' is inserted to 4. index of table 4 and 'Mehmet' is taken from the table.
---> 'Mehmet' is inserted to 6. index of table 5 and 'Insert' is taken from the table.
---> 'Insert' is inserted to 9. index of table 1 and 'CSE' is taken from the table.
---> 'CSE' is inserted to 7. index of table 2
```

These are operations in the tables, they are shown under the tables. As you can see, firstly React is inserted to table one but the cell is full, “ tables “ is taken from the cell and inserted next table until a goal cell is empty.



Insertion is successfull!

Collisions: 11	Load Factor: 1.00	Load Factor: 0.80	Load Factor: 1.00	Load Factor: 0.60	Load Factor: 0.40
indexes	**Table 1**	**Table 2**	**Table 3**	**Table 4**	**Table 5**
0	Advanced	Robin Hood	trie	7515	
1	suffix		Bu ders	Huffman	Nasılsın
2	final	polo	Kali		
3	array	dynamic	Hashing		
4	Dota 2	Rainbow six siege	Structure	Cuckoo	to go
5	React	tables	index	İyiyim	ready
6	Işık	güzel	Merhaba	ADS	Mehmet
7	hash functions	CSE	Data		
8	test	project	hash		
9	Insert		MUM	r5	

This is the tables after insertion of “React”.