



Advanced Data Structures and Algorithms

Associate Professor **Dr. Raed Ibraheem Hamed**

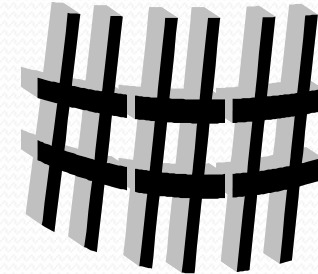
University of Human Development, College of Science and Technology
Computer Science Department

2015 – 2016



What this Lecture is about:

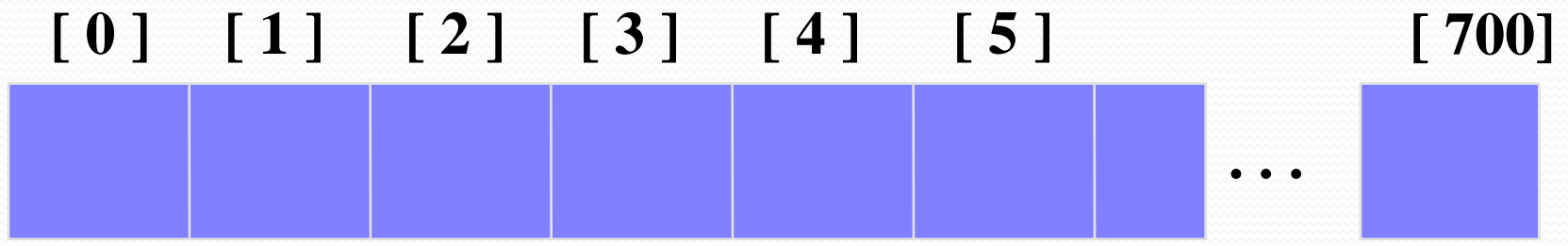
- Hash tables
- What is a Hash Table ?
- presentation of hash tables.
- Inserting a New Record
- Collisions
- Searching for a Key
- Deleting a Record
- Summary



What is a Hash Table ?

In computing, a **hash table** is a data structure used to implement an associative array, a structure that can map keys to values. A **hash table** uses a **hash** function to compute an index into an array of buckets or slots, from which the desired value can be found.

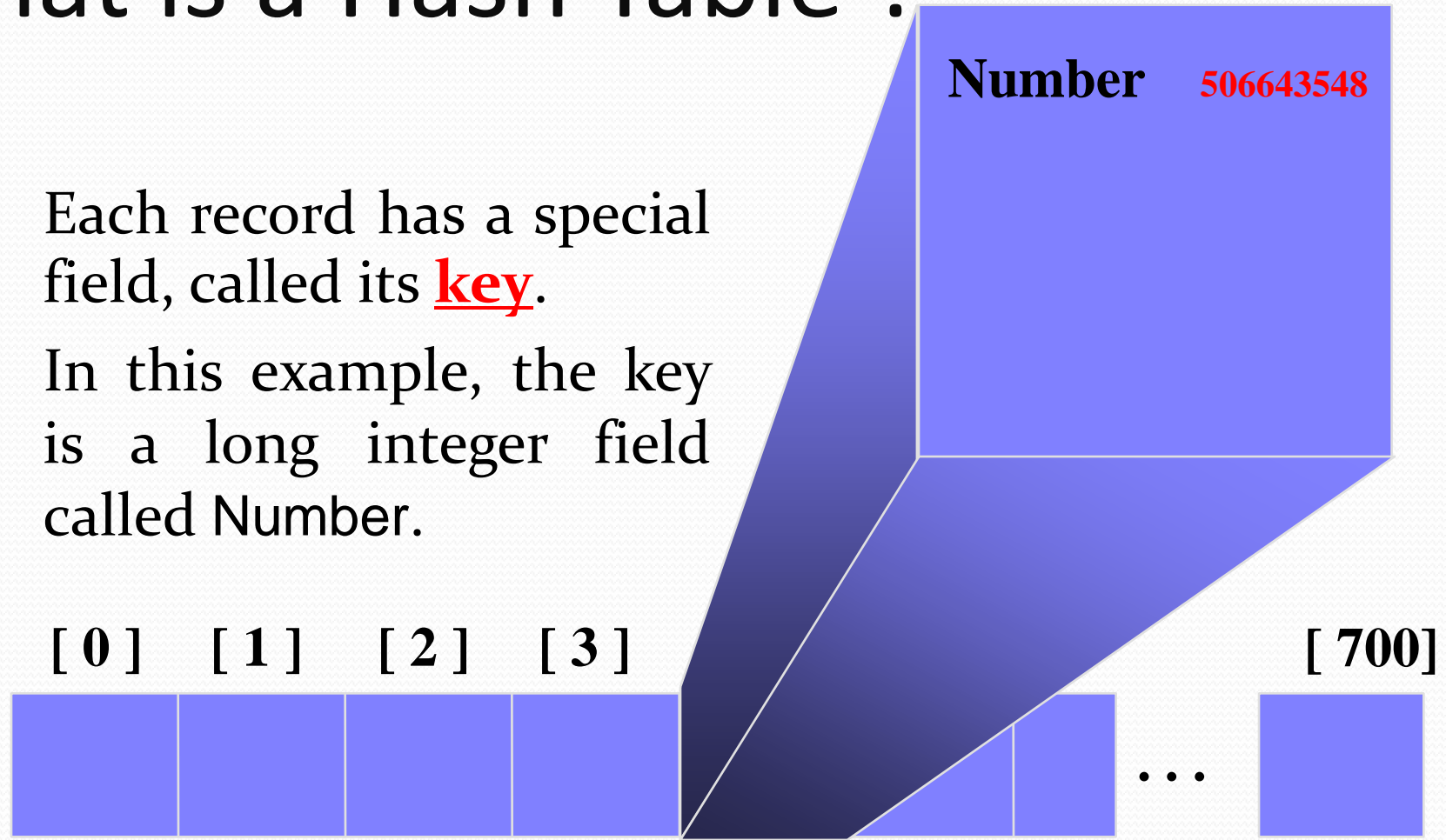
- **Hash tables** are a common approach to the storing/searching problem.
- This example has **701** records.



An array of records

What is a Hash Table ?

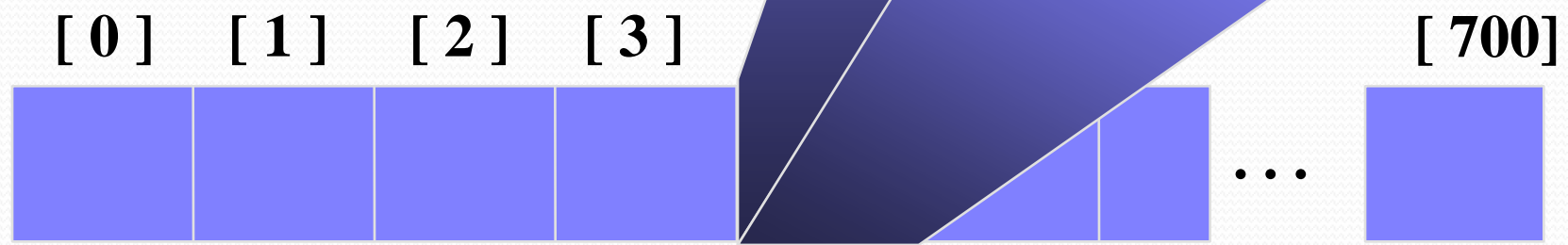
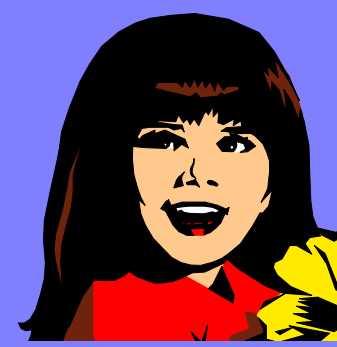
- Each record has a special field, called its **key**.
- In this example, the key is a long integer field called Number.



What is a Hash Table ?

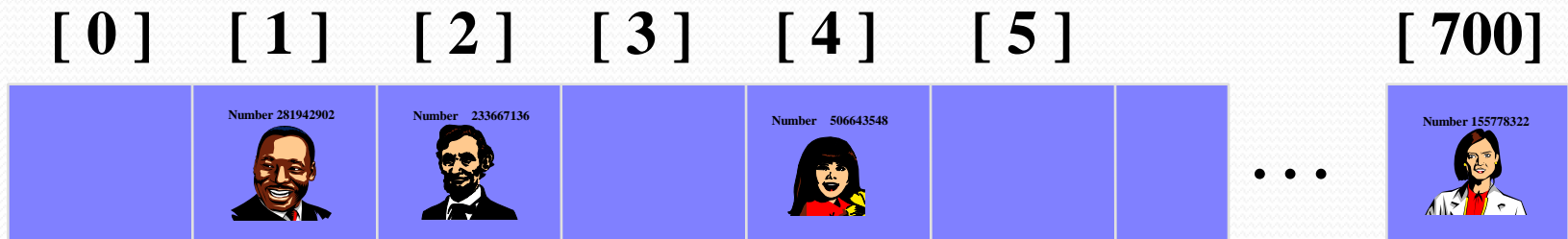
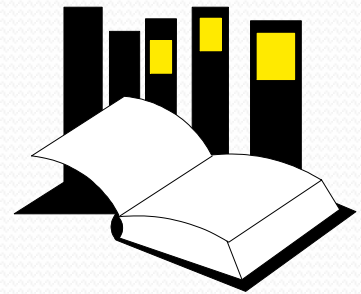
- The number might be a person's identification number, and the rest of the record has information about the person.

Number 506643548



What is a Hash Table ?

- When a hash table is in use, some spots contain valid records, and other spots are "empty".



Hash Function

One common method of determining a hash key is the **mod method** of hashing. The formula that will be used is:

hash key = key % number of slots in the table

36, 18, 72, 43, 6

the choice of hash function and table size needs to be carefully considered.

Assume a table with 8 slots:

Hash key = key % table size

$$4 = 36 \% 8$$

$$2 = 18 \% 8$$

$$0 = 72 \% 8$$

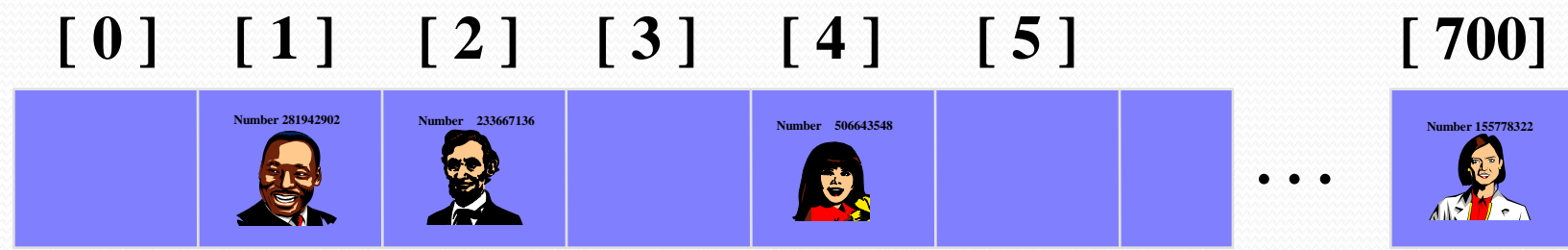
$$3 = 43 \% 8$$

$$6 = 6 \% 8$$

[0]	72
[1]	
[2]	18
[3]	43
[4]	36
[5]	
[6]	6
[7]	

Inserting a New Record

- In order to insert a new record, the **key** must somehow be **converted to** an array **index**.
- The index is called the **hash value** of the key.

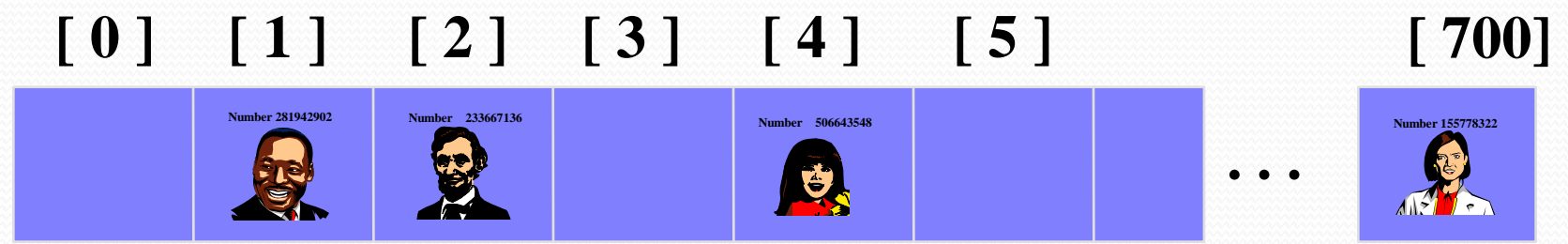


Inserting a New Record

- Typical way create a hash value:



What is $(580625685 \bmod 701)$?

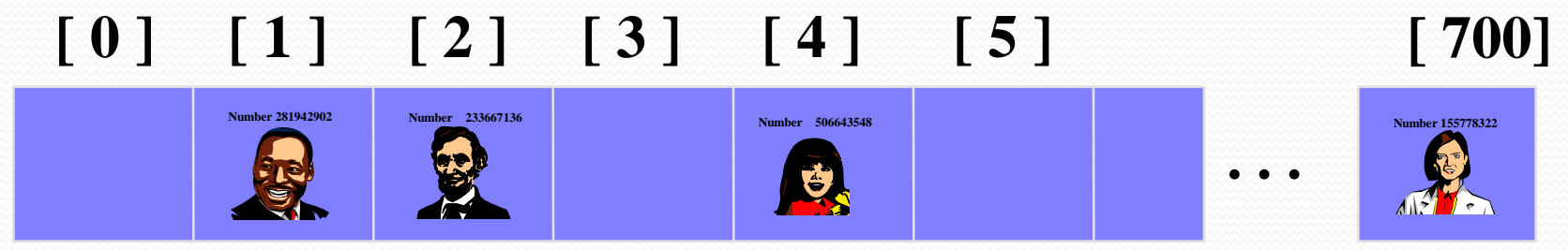


Inserting a New Record

- Typical way to create a hash value:

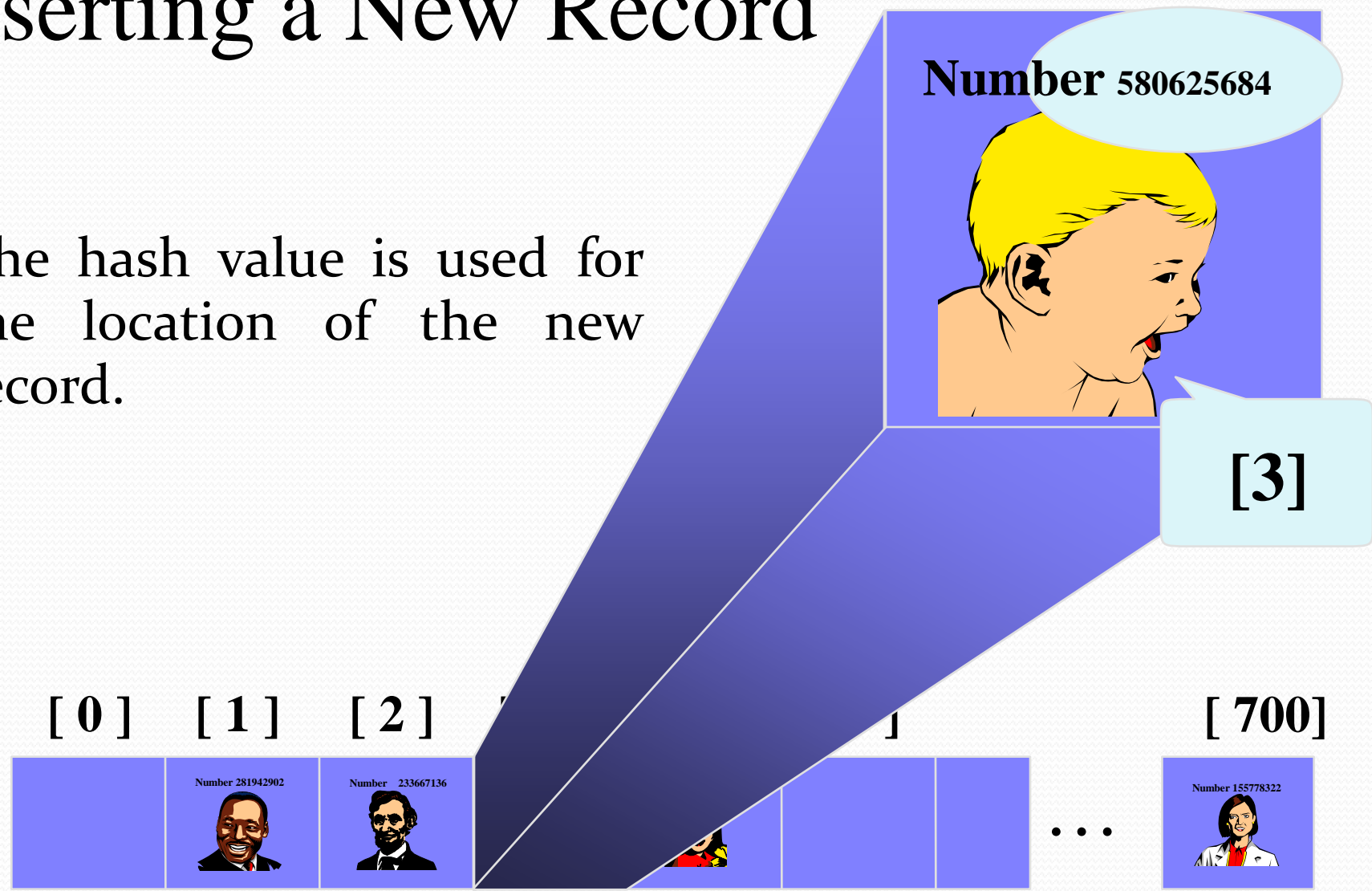


What is $(580625684 \bmod 701)$?



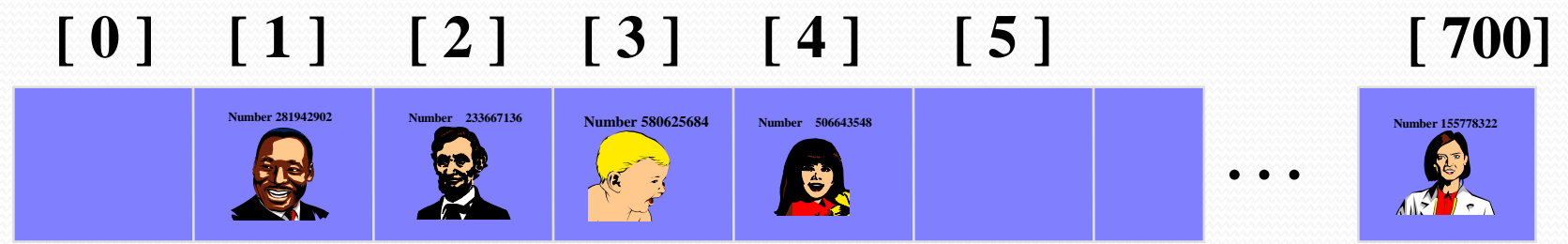
Inserting a New Record

- The hash value is used for the location of the new record.



Inserting a New Record

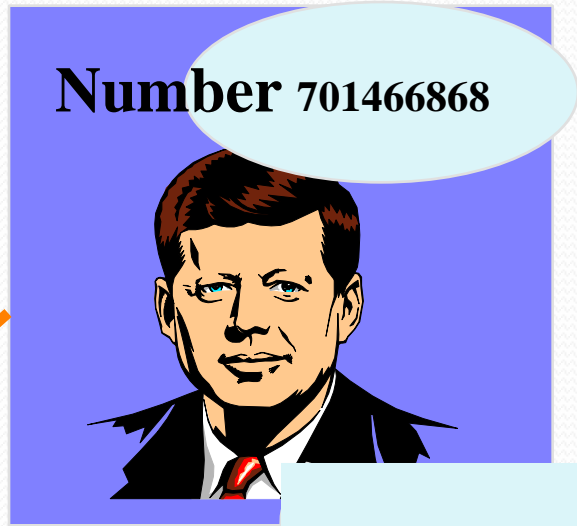
- The hash value is used for the location of the new record.



Collisions

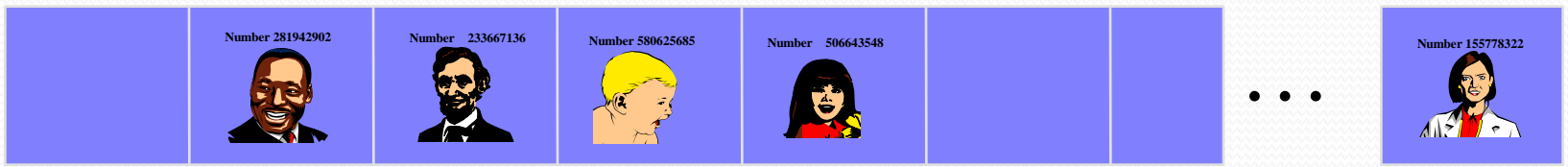
- Here is another new record to insert, with a hash value of **2**.

Sometimes, two different records might end up with the same hash value.



My hash value is **[2]**.

[0] [1] [**2**] [3] [4] [5] ... [700]

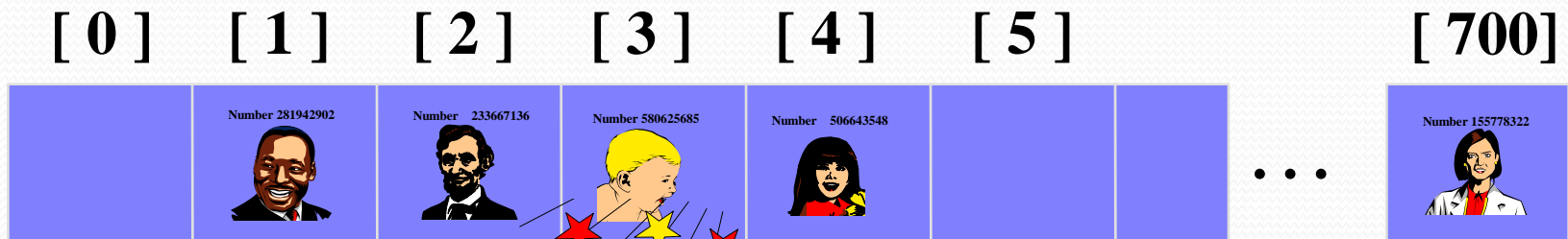


Collisions

- This is called a **collision**, because there is already another valid record at [2].



When a collision occurs, move forward until you find an empty spot.

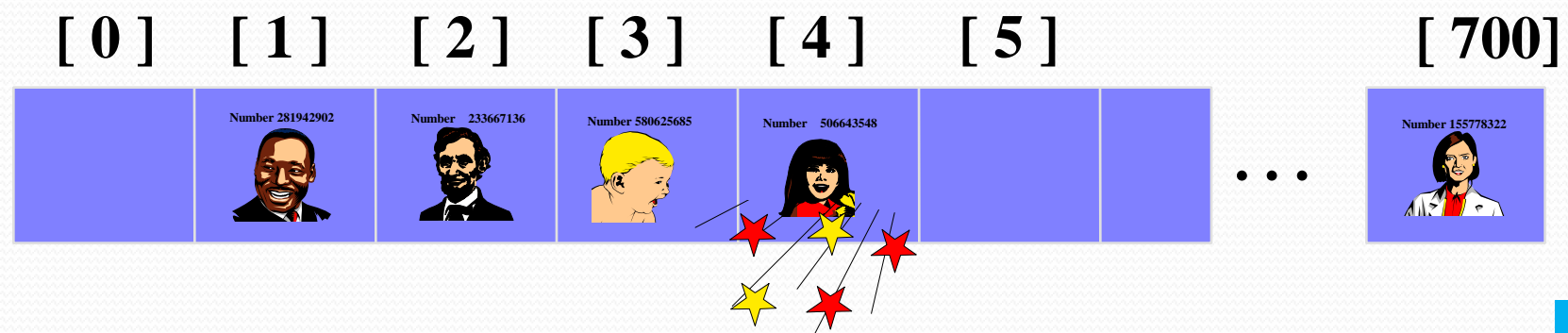


Collisions

- This is called a **collision**, because there is already another valid record at [2].



When a collision occurs, move forward until you find an empty spot.



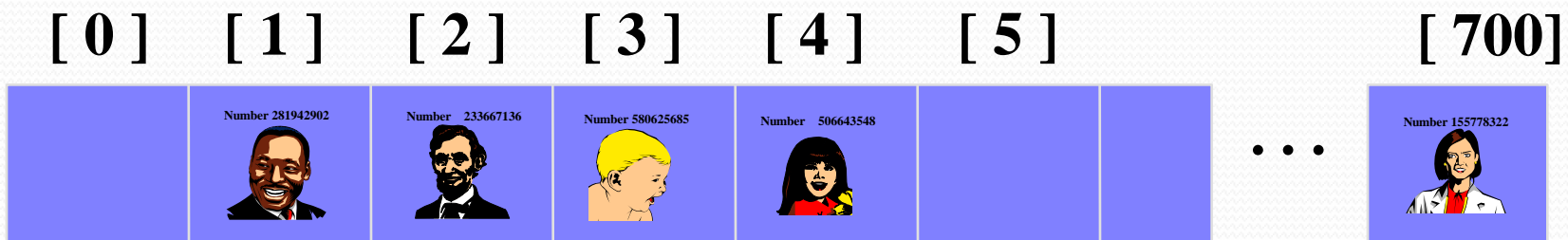
Collisions

- This is called a **collision**, because there is already another valid record at [2].

When a collision occurs, move forward until you find an empty spot.



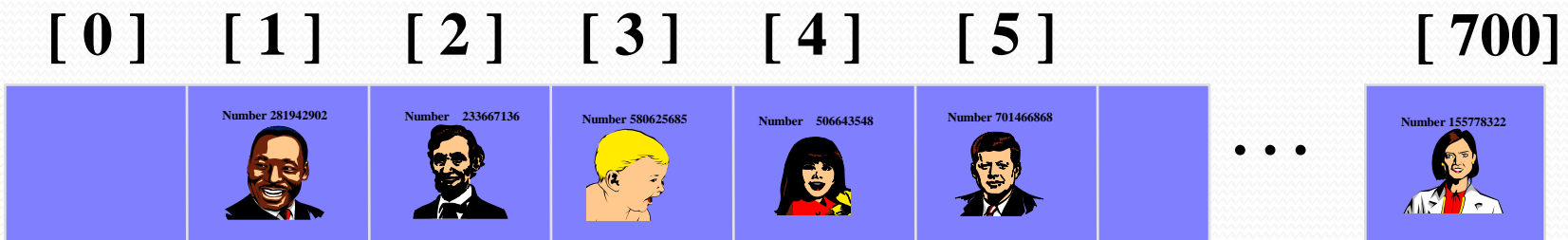
But if there are any empty spots, eventually you will reach an empty spot, and the new item is inserted here.



Collisions

- This is called a **collision**, because there is already another valid record at [2].

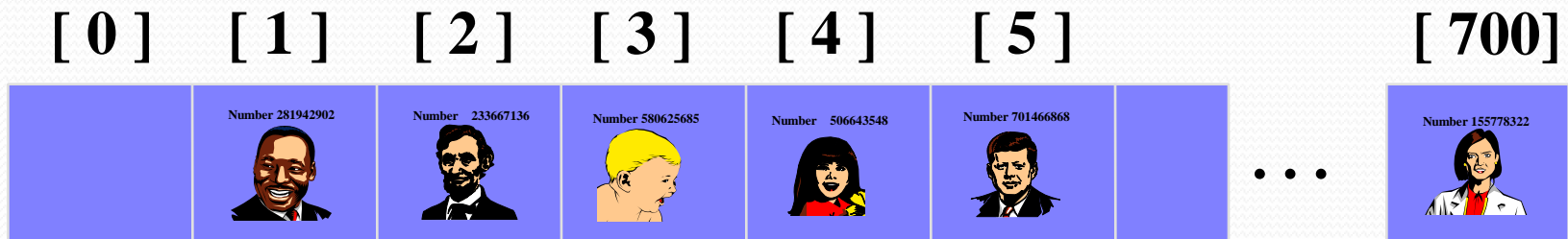
The new record goes in the empty spot.



Searching for a Key

Number 701466868

- The data that's attached to a key can be found fairly quickly.



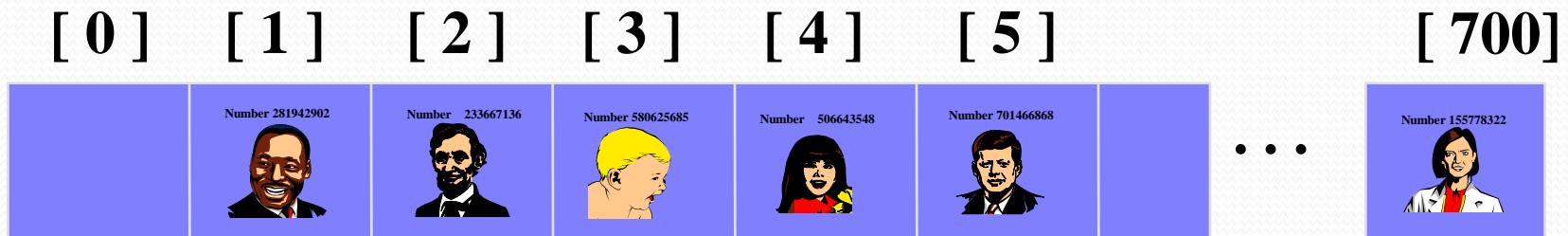
Searching for a Key

- Calculate the hash value.
- Check that location of the array for the key.

Number 701466868

My hash value is [2].

Not me.



Searching for a Key

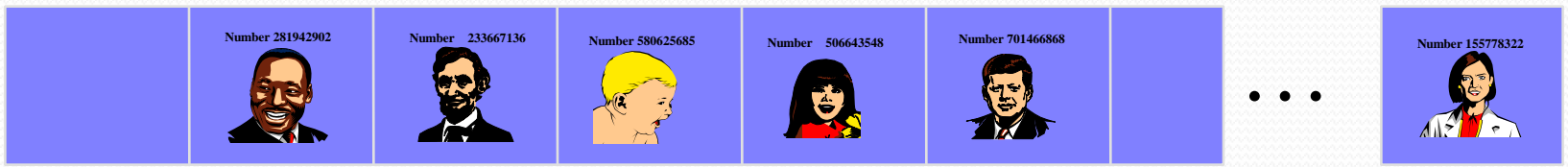
- Keep moving forward until you find the key, or you reach an empty spot.

Number 701466868

My hash value is [2].

Not me.

[0] [1] [2] [3] [4] [5] ... [700]



Searching for a Key

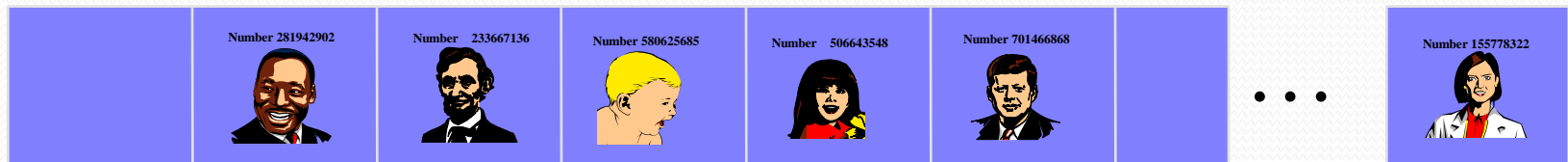
- Keep moving forward until you find the key, or you reach an empty spot.

Number 701466868

My hash value is [2].

Not me.

[0] [1] [2] [3] [4] [5] ... [700]



Searching for a Key

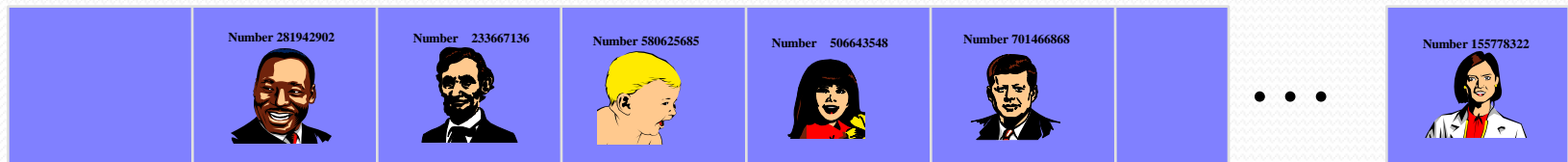
- Keep moving forward until you find the key, or you reach an empty spot.

Number 701466868

My hash value is [2].

Yes!

[0] [1] [2] [3] [4] [5] ... [700]



Searching for a Key

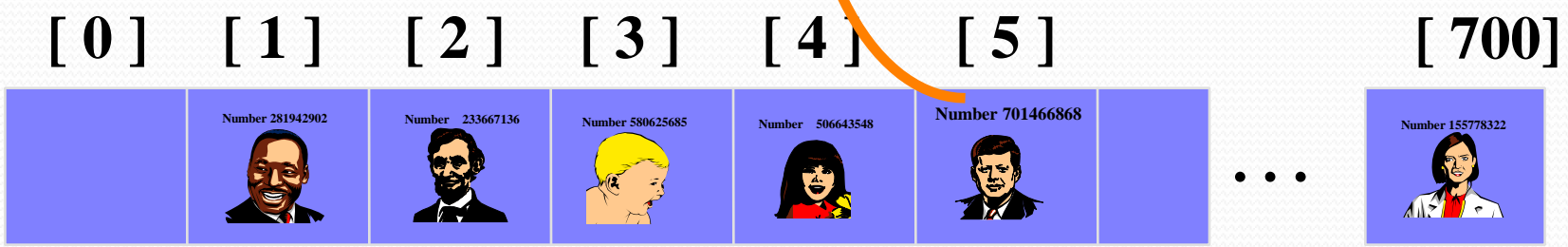
- When the item is found, the information can be copied to the necessary location.

Number 701466868



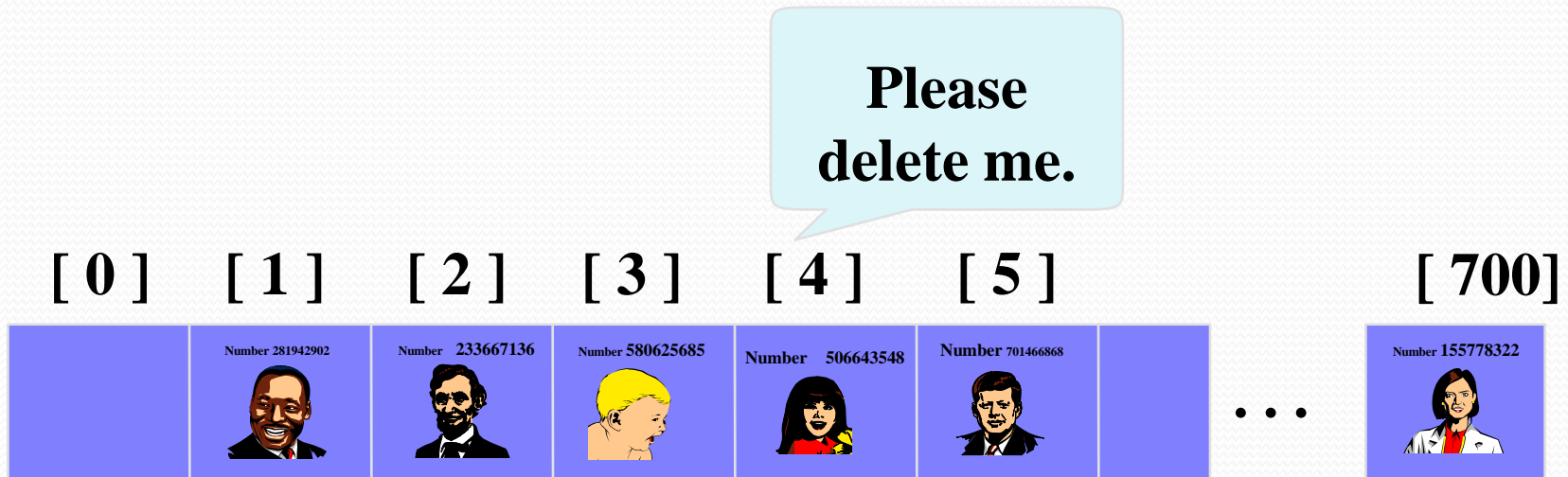
My hash value is [2].

Yes!



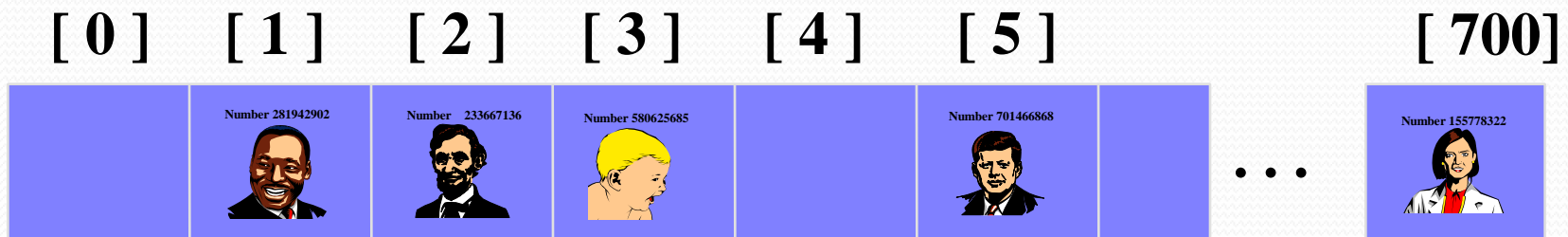
Deleting a Record

- Records may also be deleted from a hash table.



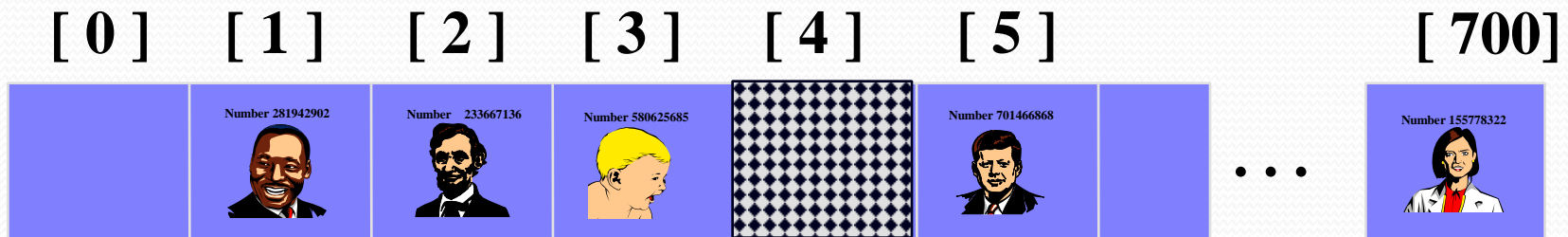
Deleting a Record

- Records may also be deleted from a hash table.
- But the location must not be left as an ordinary "empty spot" since that could interfere with searches.



Deleting a Record

- Records may also be deleted from a hash table.
- But the location must not be left as an ordinary "empty spot" since that could interfere with searches.
- The location must be marked in some special way so that a search can tell that the spot used to have something in it.





Summary

- ❑ Hash tables store a collection of records with keys.
- ❑ The location of a record depends on the hash value of the record's key.
- ❑ When a collision occurs, the next available location is used.
- ❑ Searching for a particular key is generally quick.
- ❑ When an item is deleted, the location must be marked in a special way, so that the searches know that the spot used to be used.

Thank you for listening!

???