# Time-Based Key-Value Store

Try to solve the Time-Based Key-Value Store problem.

We'll cover the following



## Statement

Implement a data structure that can store multiple values of the same key at different timestamps and retrieve the key's value at a certain timestamp.

You'll need to implement the **TimeStamp** class. This class has the following functions:

- **Init():** This function initializes the values dictionary and timestamp dictionary.
- **Set Value(key, value, timestamp):** This function stores the key and value at any given timestamp.
- **Get Value(key, timestamp):** This function returns the value set for this key at the specified timestamp.

> **Note**: When a query requests the value of a key at a timestamp that is more recent than the most recent entry for that key, our data structure should return the value corresponding to the most recent timestamp.

**Constraints:**

- $1 \leq$ `key.length`, `value.length` $\leq 100$
- `key` and `value` consist of lowercase English letters and digits.
- $1 \leq$ `timestamp` $\leq 10^3$
- At most $2 \times 10^3$ calls will be made to **Set Value** and **Get Value**.
- All the timestamps, `timestamp`, of **Set Value** are strictly increasing.

## Examples

### Sample example 1

**Input**

**Set Value:** `key = "Courses", value = "DSA", timestamp = 3`

**Get Value:** `key = "Courses", timestamp = 3`

---

**Output**

`"DSA"`

### Sample example 2

**Input**

**Set Value:** `key = "Courses", value = "OOP", timestamp = 5`

**Get Value:** `key = "Courses", timestamp = 7`

---

**Output**

`"OOP"`



## Understand the problem

Let's take a moment to make sure you've correctly understood the problem. The quiz below helps us to check that you're solving the correct problem:

Time-Based Key-Value Store

1

Given this series of commands, what is the output of the last command?

Init()

Set Value("prediction", "winning", 3)

Get Value("prediction", 2)

A) "winning"

B) "losing"

C) ""

Submit Answer

Question 1 of 3
0 attempted

Reset Quiz

## Figure it out!

We have a game for you to play. Rearrange the logical building blocks required to implement **Get Value()**.

Drag and drop the cards to rearrange them in the correct sequence.

In order to get a value, verify if the given key exists.

Verify if the timestamp we're passing to get a value is greater than the previous timestamp.

Return an empty string if the required timestamp is less than the timestamps that were set previously.

Otherwise, return the value for the respective key and timestamp.

Reset | Show Solution | Submit

## Try it yourself

Implement your solution in `TimeStamp.java` in the following coding playground. We have provided a useful code template in the other file that you may build on to solve this problem.

Java

TimeStamp.java

BinarySearch.java

```
1  import java.util.*;
2
3  class TimeStamp {
7      }
8      // Set TimeStamp data variables
9      public boolean setValue(String key, String value, int timestamp) {
10         // Write your code here
11         return false;
12     }
13     // Get the value for the given key and timestamp
14     public String getValue(String key, int timeStamp) {
15         // Write your code here
16         return "";
17     }
18 }
```

Powered by AI | Submit

Test Cases | Results

Case 1 | Case 2 | Case 3

Input #1

["TimeStamp"]

Input #2

[[]]

Time-Based Key-Value Store

Back

Solution: Snapshot Ar...

Next

Solution: Time-Based ...

Mark as Completed