CS 315 Homework Assignment 1
Turan Mert Duran
21601418
Section 2

```dart
void main() {
  var phoneList = {  //INITIALIZATION
    "Ahmet": 053245934321,
    "Ali": 05424456454,
      "Turan" : 05834323133
  };
  print(phoneList['Ahmet']); // GETTIN VALUE OF GIVEN KEY
  phoneList["Eklemeddin"] = 052319324156; // ADDING ELEMENT
  phoneList.remove("Ahmet");//removing Element
  phoneList["Eklemeddin"] = 042315534332; // MODIFYING ELEMENT
  print(phoneList.containsKey("Turan")); // SEARCH KEY
  print(phoneList.containsValue(05834323133)); // SEARCH VALUE
  foo(phoneList);

}
void foo( var phoneList) {
  phoneList.forEach((k, v) {
    print('{ key: $k, value: $v }');
  });
}
```

My dart code is firstly creating an associative array in Dart language called phoneList. My phoneList includes 3 different key-value pairs. It searches (Ahmet) value by given key and prints its' value. Then, it adds a new key-value pair Ekmeleddin (key), 053434343 (value). It removes key-value pair of Ahmet : 05343432332. It modifies value of the key Ekmeleddin. Then it checks whether the list includes any pair with key called Turan and prints result. After, it checks whether the list includes any pair with value called 05834323133 and prints result. After all, I decleared a function called foo that goes all pairs and prints key-value pairs.


```
<html>

<body>

  <script>
        // initialized ass. array
    var phoneList = { "Turan": 123, "Veli": 542, "Mustafa": 3 };
    // get value of given key
    document.write(phoneList["Turan"]);
    document.write('-----');

    //add a new element
    phoneList["Mehmet"] = 44444;
    document.write(phoneList["Mehmet"]);

    // remove an element
    delete phoneList["Mustafa"];

    //modify value of existing item
    document.write('-----');
    phoneList["Veli"] = 231453;
    document.write(phoneList["Veli"]);

    //Search for existance of key
    document.write('-----');
    document.write(phoneList.hasOwnProperty("Veli"));

    //Search for existance of value
    document.write('-----');
    function findNumber(no) {
        for (var key in phoneList) {
            if (phoneList[key] == no) return key;
        }
     return false;
    }
    document.write(findNumber(231453));
```

```
    //Prints function key-value pair with function
    document.write('-----');
    function foo() {
        for (var key in phoneList) {
            document.write(key);
            document.write(":");
            document.write(phoneList[key]);
            document.write("--");
        }
     return false;
    }
    foo();

  </script>

</body>

</html>
```

My javascript code is firstly creating an associative array in javascript language called phoneList. My phoneList includes 3 different key-value pairs. It searches (Turan) value by given key and prints its' value. Then, it adds a new key-value pair Mehmet (key), 44444 (value). It removes key-value pair of Mustafa : 3. It modifies value of the key Veli. Then it checks by hasOwnProperty method that javascript offers programmers whether the list includes any pair with key called Veli and prints result. I decleared a function called findNumber to check whether a key is in found in our phoneList array or not. In function it goes iteratively in array and checks whether there is such an key called with given name and returns result. It checks whether the list includes any pair with value called 231453 and prints result. After all, I decleared a function called foo that goes all pairs and prints key-value pairs.

```lua
local phoneList = { ["Ahmet"] = 5454, ["Turan"] = 2523}
print(phoneList["Ahmet"]) -- print value with key
phoneList["Halil"] = 3414 -- add
phoneList["Ahmet"] = nil -- remove
phoneList["Halil"] = 3423  -- modify
if phoneList["Halil"] then print("Yes it includes") end -- Search by key
    for i,v in pairs(phoneList) do
   print(i)
   print(v)
    end

    function get_key_for_value( t, value ) -- Search by value method
        for k,v in pairs(t) do
          if v==value then return k end
        end
        return nil
      end
    print(get_key_for_value(phoneList, 3423))

foo = function ()
    for i,v in pairs(phoneList) do
    io.write("Key : ")
   print(i)
    io.write("Value : ")
   print(v)
    end
  end


foo() -- Call Function Prints all keys and values
```

My Lua code is firstly creating an associative array in Lua language called phoneList. My phoneList includes 2 different key-value pairs. It searches (Ahmet) value by given key and prints its' value. Then, it adds a new key-value pair Halil (key), 3414 (value). It removes key-value pair of Ahmet : 5454. It modifies value of the key Halil. Then it checks whether the list includes any pair with key called Halil and prints result. I decleared a function called get_key_for_value to check whether a key is in found in our phoneList array or not. In function it goes iteratively in array and checks whether there is such an key called with given name and returns result. It checks whether the list includes any pair with value called 3423 and prints result. After all, I decleared a function called foo that goes all pairs and prints key-value pairs.

```php
<?php
//Initialize
$phoneList = array("Zeynep"=>"123", "Turan"=>"3347", "Musa"=>"43");
 // Get value of Given Key
echo $phoneList['Turan'] ;
echo "\n ";

// Add a new element
$phoneList['Halil'] = "3437";
//Remove an element
unset($phoneList["Turan"]);
//Modify the value of an existing element
$phoneList['Halil'] = "3333";
//Search for the existence of a key
if (array_key_exists("Halil",$phoneList))
  {
  echo "Key is available!\n ";
  }
else
  {
  echo "Key is not available!\n ";
  }

//Search for the existence of a value
if (in_array("43", $phoneList))
  {
  echo "found\n ";
  }
else
  {
  echo "not found\n ";
  }
function foo($list) {
// Print key-value pair with function
  foreach($list as $key => $value) {
    echo "Key=" . $key . ", Value=" . $value;
    echo "\n";
  }
}
foo($phoneList);
```

My PHP code is firstly creating an associative array in PHP language called phoneList. My phoneList includes 3 different key-value pairs. It searches (Turan) value by given key and prints its' value. Then, it adds a new key-value pair Halil (key), 3437 (value). It removes key-value pair of Turan : 3347. It modifies value of Halil as 3333.By in_array method that PHP offers programmers my code checks whether there is such an key called with given name and returns result. It checks whether the list includes any pair with value called 43 and prints result. After all, I decleared a function called foo that goes all pairs and prints key-value pairs iteratively.

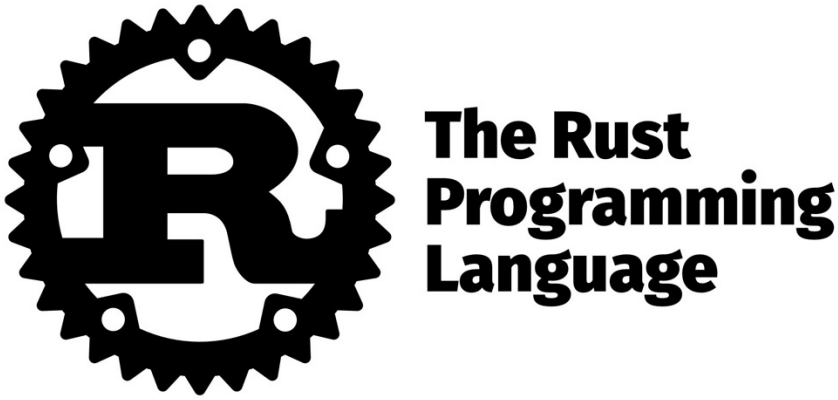

```
print('Hello World!')
#Initialize array
phoneList = {
  "Turan": 1234,
  "Veli": 3342,
  "Mustafa": 343255
}
# Get value of given key
print(phoneList["Turan"])
# Add a new element
phoneList["Ahmet"] = 53434
#Remove an element
phoneList.pop("Turan")
#Modify value of existing item
phoneList["Veli"] = 3341
#Search for existance of key
print(phoneList.get("Veli"))
#Search for existance of value
print(343255 in phoneList.values() )
# Function prints the key-value pair
def foo():
  for x in phoneList:
    print(x, " - ", phoneList[x] )

foo()
```

My Python code is firstly creating an associative array in Python language called phoneList. My phoneList includes 3 different key-value pairs. It searches (Turan) value by given key and prints its' value. Then, it adds a new key-value pair Ahmet (key), 53434 (value). It removes key-value pair of Turan : 1234. It modifies value of the key Veli. Then it checks whether the list includes any pair with key called Veli by xxx.get method that Python offers programmers and prints result.Then it checks whether the list includes any pair with value called 343255 and prints result. After all, I decleared a function called foo that goes all pairs and prints key-value pairs iteratively.


```ruby
#Inıalize
phoneList = {'Turan' => 1234, 'Ahmet' => 5453, 'Zeynep' => 19434}
#Get value for given key
puts phoneList['Turan']
#Add a new element
phoneList['Veli'] = 452
#Remove a element
phoneList.delete('Turan')
#Modify the value of an existing element
phoneList['Veli'] = 451
#Search for the existence of a key
puts phoneList.key?('Veli')
#Search for the existence of a value
puts phoneList.value?(5453)
#function prints the key-value pair
def foo(h)
  h.each_pair do |k,v|
    if v.is_a?(Hash)
      ihash(v)
    else
      puts "#{k} - #{v}"
    end
  end
end
foo(phoneList)
```

My Ruby code is firstly creating an associative array in Ruby language called phoneList. My phoneList includes 3 different key-value pairs. It searches (Turan) value by given key and prints its' value. Then, it adds a new key-value pair Veli (key), 452 (value). It removes key-value pair of Turan : 1234. It modifies value of the key Veli. Then it checks whether the list includes any pair with key called Veli by xxx.key?("key") method that Ruby offers programmers and prints result. Then it.checks whether the list includes any pair with value called 5453 and prints result by xxx.value?("value") method that Ruby offers. After all, I decleared a function called foo that goes all pairs and prints key-value pairs iteratively.




```rust
use std::collections::HashMap;

fn main() {
    // Initializing
    let mut phoneList = HashMap::new();
    phoneList.insert(String::from("Turan"), 13242);
    phoneList.insert(String::from("Ahmet"), 3454);
    phoneList.insert(String::from("Radman"), 2423);
    phoneList.insert(String::from("Gunes"), 430);
    // Getting Value with key
    println!("{}", phoneList["Turan"]);
    // Add a new element
    phoneList.insert(String::from("Zeynep"), 431);
    // Remove an element
    phoneList.remove("Turan");
    // Modify value
    *phoneList.get_mut("Gunes").unwrap() = 432;
    println!("{}", phoneList["Gunes"]);
    //Search for the existence of a key
    println!("{}", phoneList.contains_key("Gunes"));
    //Search for the existence of a value
    let mut searchedValue : i32;
    searchedValue = 2423;
    checkValue(&mut phoneList, &mut searchedValue);
    //println!("{}", phoneList.containsValue(430));
    printAll(&mut phoneList);
```

```
}
    fn printAll(phoneList: &mut HashMap<String, i32> ){
        for (key, value) in phoneList.iter() {
        println!("{} - {}", key, value);
      }
    }
    fn checkValue(phoneList: &mut HashMap<String, i32>, no: &mut i32){
        for (key, value) in phoneList.iter() {
        if no == value{
          println!("Value found in List!");
        }
      }
    }
```

My Rust code is firstly creating an associative array in Rust language called phoneList by using HashMap that Rust offers programmers. My phoneList does not include key-value pairs. In order to fill it I used insert method of HashMap. Then after all inserting key-value instructions, It searches (Turan) value by given key and prints its' value. Then, it adds a new key-value pair Zeynep (key), 431 (value). It removes key-value pair of Turan : 13242. It modifies value of the key Gunes. Then it checks whether the list includes any pair with key called Gunes by xxx.contains_key("key") method that Ruby offers programmers and prints result. Then it.checks whether the list includes any pair with value called 2423 and prints result by a function called checkValue that i have created (go iteratively all the list and prints value found if key-value pair is available). After all, I decleared a function called foo that goes all pairs and prints key-value pairs iteratively.

Evaluation of these languages in terms of readability and writability of associative array data structure

In terms of readability and writability Dart programming language was easy to write and read. Implementation of associative array was easy and understandable not too complex. It has all methods that programmer would need to use (. remove, .containsKey, .containsValue).By just looking example codes of Dart, I understand its' syntax and started to write code. I think it would be best programming language among others. Because in other languages i forced to write functions for check contains value that took my time. However, in Dart these functions were ready to use. The worst of among these languages in terms of readability and writability was the Rust programming language i think. I was forced to write

how many byte is needed for integers, write function for checking key by value. Moreover, iterating array is a little bit different than other programming languages and weird I thinkş. I couldn't understand anything about Rust by just knowing nothing about Rust language and skimming example codes.

My Learning Strategy

My learning strategy was not too complex. Firstly, I used Google for searching example codes of languages. I skimmed them. After each skimming, I had a little informations about how to write this languages. Then after skimming, I used again Google to search about associative arrays .Thus, I remembered what associative arrays were. I used w3schools.com for learning associative arrays. Then I returned back to the Google and started searching about associative arrays in each programming languages that I need to write.All websites that I used for learning can be found below this part. I used VSCode to write all languages, for running them I used online compilers and Google Chrome for html.

Compilers / Interpreters:

- https://repl.it for (Lua, Rubby, PHP, Rust, Python) *(Acc. Date: 26.11.2020)*
- https://dartpad.dev for (Dart) *(Acc. Date: 26.11.2020)*
- Google Chrome (for JS&HTML)

Informative websites:

- Ruby Learning (http://rubylearning.com/satishtalim/ruby_hashes.html) *(Acc. Date: 26.11.2020)*
- stackoverflow (https://stackoverflow.com/questions/9279768/how-do-i-loop-over-a-hash-of-hashes-in-ruby) *(Acc. Date: 26.11.2020)*
- W3schools (https://www.w3schools.com/python/python_dictionaries_loop.asp) *(Acc. Date: 26.11.2020)*
- Tutorial Republic (https://www.tutorialrepublic.com/faq/how-to-delete-an-element-from-an-array-in-php.php) *(Acc. Date: 26.11.2020)*

- BezKoder
  (https://bezkoder.com/dart-map/)
- Programming Idioms
  (https://programming-idioms.org/idiom/8/initialize-a-new-map-associative-array/202/dart) *(Acc. Date: 26.11.2020)*