create ():AArray

Purpose: to create an empty , initialized associative array
Preconditions: none
Postconditions: an empty , initialized associative array is created

```
create ():AArray
1. create the storage ADT
2. create the key ADT
3. return(the AArray we just created)
```

destroy(AArray)

Purpose: to destroy an associative array and free the memory if required
Preconditions: an initialized associative array
Postconditions: the array is destroyed along with all references to data. (Important note:
the data itself may not be destroyed by this operation and should be removed prior to
destroying the AArray).

```
destroy(AArray)
1. the appropriate destroy procedure
2. release any other pointers
```

insert(AArray , key , value)

Purpose: To add a key/value pair to the associative array
Preconditions: no value exists for the key being added
Postconditions: the size of the AArray has increased by one , the key and value are stored
with a reference leading from key to value.

```
insert(AArray ,key ,value)
1. key into key ADT
2. connect key and value with reference (pointer)
3. increase length counter of AArray by one
```

remove (AArray ,key): value

Purpose: to remove a key/value pair from the associative array
Preconditions: the key/value pair is stored in the AArray
Postconditions: the key/value pair is no longer in the AArray. The length of the AArray has
decreased by one.

```
remove (AArray ,key): value
1. look up key in key ADT
2. follow reference to value in value ADT
3. store value in temporary variable
```

```
4. remove value from value ADT
5. remove key from key ADT
6. return(temporary variable)
```

lookup (AArray , key): value

Purpose: to retrieve the value stored for a particular key
Preconditions: the key/value pair is in the AArray
Postconditions: the value is returned. The AArray is unchanged.

```
lookup (AArray ,key): value
1. look up key in key ADT
2. follow reference to value in value ADT
3. store value in temporary variable
4. return(temporary variable)
```

update (AArray , key , newValue)

Purpose: the change the value associated with a key that already exists in the associative array
Preconditions: a key/value pair exists for the given key
Postconditions: the new value is associated with the given key

```
1. look up key in key ADT
2. follow reference to value in value ADT
3. set value to newValue
```

exists(key):Boolean

Purpose: to determine if a key is represented in the associative array
Preconditions: an initialized AArray is available
Postconditions: n/a

```
exists(key):Boolean
1. look up key in key ADT
2. if key is found return(true)
3. else return(false)
```

isEmpty ():Boolean

Purpose: to determine if any key/value pairs are represented in the associative array
Preconditions: an initialized AArray is available
Postconditions: n/a

```
isEmpty ():Boolean
1. if the key ADT is empty; return(true)
```

```
2. else return (false)
```

isFull ():Boolean
Purpose: to determine if the AArray is full
Preconditions: an initialized AArray is available
Postconditions: n/a

```
isFull ():Boolean
1. if the key ADT is e
```