

# CS315

**Programming Languages**

**Fall 2019 Homework 3**

**Muhammad Arham Khan**

**21701848**

**Section 3**

**27 Dec, 2019**

# BRIEFING

## PROGRAM:

```
;Function definition for Pair-up
;-----

;Pair up function that takes two lists in input
;and returns an associative list
(define (pair-up listKeys listValues)
  (cond
    ;assuming that both lists are equal in size
    ;if listKeys is null, listValues is null too
    ;return null if input list is null
    ((null? listKeys) '())

    ;If listKeys is not a list and a single element
    ;listValues is single element too, so return a single
    ;pair of the two input elements (atoms)
    ;This also applies to the end case of the recursion when
    ;only one element is left in the list
    ((not (list? listKeys)) (cons (cons listKeys listValues) '()))
  )

;If the input array is a proper array
(else
  ;Merge the current pair and the associative list returned
  ;by the further recursive calls to the function
  (cons

    ;Make a pair from the first elements from both lists
    (cons (car listKeys) (car listValues))

    ;Tail recursive function call to get further paired
    ;elements for the associative list
    (pair-up (cdr listKeys) (cdr listValues))
  )
)
```

```
)
))

; Defining Associative Arrays
;-----

;Creating a function that makes an
;associative list from two elements(atoms)
(define list_test_1 (pair-up 'Arham 'Khan))
(display list_test_1)
(display "\n")

;Creating a function that makes an
;associative list from null elements
;and returns a null list
(define list_test_2 (pair-up '() '()))
(display list_test_2)
(display "\n")

;Creating a function that makes an
;associative list from two lists
(define list_test_3 (pair-up '(Queen Taylor Cardi Billie Camila) '(Herby Swift B Eilish
Cabello)))
(display list_test_3)
(display "\n")

;Creating a function that makes an
;associative list from two lists
(define list_test_4 (pair-up '(A B C D) (list 1 2 3 4)))
(display list_test_4)
(display "\n\n_____ \n\n")

;Validating Associative Arrays
;-----

;Testing for the existence of Arham in
;list returned by list_test_3
;which doesn't contain "Arham: key so
;returns #f
```

```

(display (assq 'Arham list_test_3))

(display "\n")

;Testing for the existence of Arham in
;list returned by list_test_1
;returns the pair
(display (assq 'Arham list_test_1))

(display "\n")

;Testing for the existence of Arham in
;list returned by list_test_3
;returns the pair with mentioned key
(display (assq 'Billie list_test_3))

(display "\n")

;Testing for the existence of Arham in
;list returned by list_test_4
;returns the pair
(display (assq 'B list_test_4))

(display "\n")

```

## OUTPUT:

```

;Constructing an associative list from two atoms
((Arham . Khan))

;Constructing a null associative list from null elements
()

;Constructing an associative list from two lists of 5 elements each
(((Qveen . Herby) (Taylor . Swift) (Cardi . B) (Billie . Eilish) (Camila . Cabello)))

;Constructing an associative list from one list of characters and one integer list.
((A . 1) (B . 2) (C . 3) (D . 4))



---



;Finding a key that does not exist in the associative list
#f

;Finding key Arham returns the dotted pair for that key
(Arham . Khan)

;Finding key Billie returns the dotted pair for that key
(Billie . Eilish)

;Finding key B returns the dotted pair for that key

```

**ANALYSIS:**

During the first four lines of the output, four associative lists are constructed and output. The first list constructed uses two elements(atoms) to return an associative list containing one pair. In second, we pass two null parameters so the pair-up function returns a null list and in the other two, we make two lists. During the last four lines, we try using the assq command on the lists to try finding elements using the keys and see if a dotted pair with that key exists. In the first line, we try searching for "Arham" which does not exist in the associative list. In the second list, a pair with "Arham" as the key exists so a pair containing "Arham" and "Khan". And in line 3 and 4 we search for elements that exist in the list using assq command and a pair is returned.

## **References**

[1] "Wikibooks: Scheme Programming"

[https://en.wikibooks.org/wiki/Scheme\\_Programming/List\\_Operations](https://en.wikibooks.org/wiki/Scheme_Programming/List_Operations) [Accessed Dec 27, 2019]

[2] "Scheme Expressions"

[https://www.cs.cmu.edu/Groups/AI/html/r4rs/r4rs\\_6.html](https://www.cs.cmu.edu/Groups/AI/html/r4rs/r4rs_6.html) [Accessed Dec 27, 2019]

[3] "The scheme programming language"

<https://www.scheme.com/tspl3/> [Accessed Dec 27, 2019]