



2019-2020 Fall

CS 315

Homework 3

Section: 3

Name: Munib Emre

Surname: Sevilgen

Student ID: 21602416

Scheme Language

The program:

```
(define (pair-up keys values)
  (cond
    ((null? keys) '())
    ((null? values) '())
    (else (cons
            (cons (car keys) (car values))
            (pair-up (cdr keys) (cdr values))))))
```

The pair-up function takes two lists as arguments. The keys list is the keys of the associations and the values list is the values of the associations. The function first checks whether the arguments are null or not. If at least one of them is null then it returns an empty list. If they are not null, it gets the first values of the lists and makes them pairs. It makes the consecutive pairs the pairs. Then it calls itself by the next values in the lists as arguments. However, because of the last value is always an empty list, then the pairs of the pairs becomes a list. Therefore, it is a tail-recursive function.

Sample executions:

The example at the homework page.

```
(pair-up '(a b c) (list 1 2 3))
((a . 1) (b . 2) (c . 3))
```

The data is got from the homework page association example.

```
(pair-up '(Ali Ayse Ahmet Fatma) '(windows mac linux unix))
((Ali . windows) (Ayse . mac) (Ahmet . linux) (Fatma . unix))
```

The example of the empty lists.

```
(pair-up '() '())
()
```

The example is constructed to show the function is works properly.

```
(pair-up '(mocha americano flatwhite turkish) '(7 5 6 10))
((mocha . 7) (americano . 5) (flatwhite . 6) (turkish . 10))
```