# CSI 2120 – Lab Task 4

**Abdorrahim Bahrami**

## Vectors, Loops


Your task in this lab is to get yourself familiarized with Loops and Vectors in Scheme and do some simple recursive programming. Make sure you have DrRacket installed, and you are familiar with the interpreter and the coding parts. If you need help, ask your TA to help you with this.

Then, you should do the following programming tasks. Each programming task in the lab is a design based on the subjects you learned during lectures. There are test samples that you can use to test your program. If you have questions, ask your TAs.

**Important Note: Using ChatGPT during the lab time is strictly forbidden, if your TA catch you using it, you will receive 0 for all lab tasks immediately. During lab time, you can only use online references mentioned in class.**

Task 1:

In this task, you need to write a function that takes a vector and two indices, which are the beginning and the end of a range of elements in the vector. Your function should return the sum of elements in the vector in the given range. Your function must be non-recursive. You must use loop. You can assume that indices are always valid.

```
(define (sum-sequence v begin end)

)
```

Here are some test cases to test your function.

(sum-sequence ‘#(2 6 -3 4 7 1 5) 2 5) ==> 9

(sum-sequence ‘#(6 7 4) 2 1) ==> 0

Task 2:

In this task, write a function that takes a sorted vector and an element and returns the index of that element using binary search. You probably need a helper function and that function is recursive. The function should return -1 if the given element is not in the vector. You can assume that the vector is always sorted.

(define (binary-search-recursive v x)

)

Here are some test cases to test your function.

(binary-search-recursive '#(5 12 21 34 56 67 89) 56)  ==> 4

(binary-search-recursive '#(5 12 21 34 56 67 89) 44)  ==> -1

Task 3 (Challenge): Repeat the previous task but this time, your function must be non-recursive. You must use loop for doing a binary search.

(define (binary-search-loop v x)

)