Lab 2

```
import array

class Vector():
    def __init__(self, type: str, objects: list):
        self.type = type
        self.vector = array.array(type, objects)
    def length(self):
        return len(self.vector)
    def is_valid_index(self, index):
        if index < self.length():
            return True
        return False

    def contains(self, item):
        "returns true if item is in this vector"
        for i in range(self.length()):
            if self.vector[i] == item:
                return True
        return False

    def get_item(self, index):
        if index < self.length():
            return self.vector[index]
        return None

    def set_item(self, index, item):
        if self.is_valid_index(index):
            self.vector[index] = item

    def append(self,item):
        self.vector.append(item)

    def remove(self, index):
        if self.is_valid_index(index):
            self.vector.remove(index)

    def indexOf(self, item):
        for i in range(self.length()):
            if self.get_item(i) == item:
                return i
        return None

    def extend(self, vector):
        for i in range(0, vector.length()):
            self.vector.append(vector.get_item(i))

    def subVector(self, start, end):
        if not self.is_valid_index(start) or not
self.is_valid_index(end):
            return None
        if start > end:
            return None
        else:
            sv = Vector(self.type, [])
            for i in range(start, end):
                sv.append(self.vector[i])
            return sv

    def to_string(self):
        out="<"
        for i in range(self.length()):
            out+=str(self.get_item(i))
            out+=", "

        return out[:len(out)-2] + ">"


v = Vector('i', [1,2])
v.append(3)
v.remove(2)

v.extend(Vector('i', [20,30,40,50,60,90249]))
v.set_item(0, 123)
print(v.to_string())

sv = v.subVector(3,5)
print(sv.to_string())
```