Problem A:

Write a function named $get\_inversions$ which gets an array A of integers and returns the number of inversions in the Array.
An inversion is a pair of integers i, j which $i < j$ and $A_i > A_j$.

Example:

```
> (get_inversions '(1 2 3 4))
0
> (get_inversions '(4 3 1 5))
3
```

Problem B:

Let's define an environment Env a mapping from variables to values.
Consider a list of pairs in which each pair's first element is a string (variable's name) and second element is an integer (variable's value) as a presentation of an environment in racket language.
For example consider an environment E a mapping from A to 2 and from B to 3, following list is the presentation of E:

```
'(("A" . 2) ("B" . 3))
```

Write a function named $diff\_envs$ which gets two environments $E1$ and $E2$ and returns a list of keys which two environments have different values for.
Note that both environments should include the different keys but with not the same values.

Example:

```
> (list (cons "A" 2) (cons "B" 1) (cons "C" 2))
'(("A" . 2) ("B" . 1) ("C" . 2))
> (list (cons "B" 1) (cons "C" 4))
'(("B" . 1) ("C" . 4))
> (diff_envs (list (cons "A" 2) (cons "B" 1) (cons "C" 2)) (list
(cons "B" 1) (cons "C" 4)))
'("C")
```

Problem C:

A bracket sequence is called regular if it is possible to obtain correct arithmetic expression by inserting characters “+” and “1” into this sequence. For example, sequences “(())()”, “()” and “(()(()))” are regular, while “)(“, “(()” and “(()))(“ are not.

Write a function named $generate\_all$ which gets a number N and returns list of all regular bracket sequence of length $2 \times N$.

Example:

```
> (generate_all 1)
'("()")
> (generate_all 2)
'("(())" "()()")
```