**Problem 1**

Given a string s, find the longest palindromic substring in s. You may assume that the maximum length of s is 1000.

Example 1:

Input: "babad"

Output: "bab"

Note: "aba" is also a valid answer.

Example 2:

Input: "cbbd"

Output: "bb"

Define a helper function that either returns true or false if the given substring is a palindrome

Break down mainstream in substrings. Two loops. Outer loop grab first letter of substring. Inner loop index use to make a slice of string. substring.slice(0, 5). If is\_palindrome?(substring). If returns true, add it to longest palindrome variable if current substring is greater than current longest palindrome.

Function is\_palindrome?(substring) {

Substring === substring.reverse()

}

Function longest\_pal(string) {

Let longest = “”;

Let chars = string.split(“”);

for(let i = 0; i < chars.length; i++) {

for(let j = 1; j < chars.length; j++) {

Let substring = chars.slice(i,j+ 1)

If (is\_palindrome?(substring) && substring.length > longest.length) {

Longest = substring;

}

}

Return longest;

}

}

**Problem 2**

Write an efficient algorithm that searches for a value in an m x n matrix. This matrix has the following properties:

Integers in each row are sorted from left to right.

The first integer of each row is greater than the last integer of the previous row.

Example 1:

Input:

matrix = [

[1, 3, 5, 7],

[10, 11, 16, 20],

[23, 30, 34, 50]

]

target = 3

Output: true

Example 2:

Input:

matrix = [

[1, 3, 5, 7],

[10, 11, 16, 20],

[23, 30, 34, 50]

]

target = 13

Output: false

Check the last value of each row, if the target === the last item, return it, or the first row found where the target is less than one of last items, only search that row, starting from the next to last, since we’ve already searched the end.

Function findTarget(matrix, target){

AmountOfRows = matrix.length - 1;

For (let i = 0; i < amountOfRows - 1; i++){

For (let j = matrix[i].length - 1; j > 0; j-- ){ //[1, 3, 5, 7],

If (matrix[i][j] === target){

return true;

} else

}

}

}