Please upload your solutions into github and me the link. Thank you.

**\*\*Problem 1:\*\***

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

@Component

class LongestPalin {

private int lo, maxLen;

public String longestPalindrome(String s) {

int len = s.length();

if (len < 2)

return s;

for (int i = 0; i < len-1; i++) {

extendPalindrome(s, i, i);

extendPalindrome(s, i, i+1);

}

return s.substring(lo, lo + maxLen);

}

private void extendPalindrome(String s, int j, int k) {

while (j >= 0 && k < s.length() && s.charAt(j) == s.charAt(k)) {

j--;

k++;

}

if (maxLen < k - j - 1) {

lo = j + 1;

maxLen = k - j - 1;

}

}

}

**\*\*Solution?:\*\***

Write a Spring Boot micro-service that completes the following:

1. Receives string input over an API

2. Stores the longest palindrome in provided database

3. Retrieves the stored palindrome over an API

You should be able to run your micro-service locally and use an API client (such as PostMan) to test your results.

@RestController

class Controller {

@Autowired

private LongestPalin longestPalin;

@Autowired

private Service service;

@RequestMapping(value = “/palin”, method = RequstMethod.POST)

public void save(String input) {

String res = longestPalin.longestPalindrome(input);

service.save(res);

}

@RequestMapping(value = “/palin”, method = RequstMethod.GET)

public String save(String input) {

String res = service.getByInput(input);

If(res != null) {

return res;

}

}

}

**Problem 2/Solution?:**

Have the function BinaryReversal(str) take the str parameter being passed, which will be a positive integer, take its binary representation (padded to the nearest N \* 8 bits), reverse that string of bits, and then finally return the new reversed string in decimal form. For example: if str is "47" then the binary version of this integer is 101111 but we pad it to be 00101111. Your program should reverse this binary string which then becomes: 11110100 and then finally return the decimal version of this string, which is 244.

public int BinaryReversal(str) {

int highest = 0;

int temp = str;

String bin = “”;

while(temp > 0) {

bin = (temp & 1) + bin;

highest++;

temp >>= 1;

}

int count = highest % 2 == 0? highest / 2 : highest / 2 + 1;

for(int i = 0; i < count\*8 – highest; i++) {

bin = “0” + bin;

}

Int left = 0, right = bin.length() – 1;

char[] chars = bin.toCharArray();

while(left < right) {

char cur = left;

chars[left] = chars[right];

chars[right] = cur;

left++;

right--;

}

int res = 0;

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

res += (chars[i] – ‘0’) << (chars.length - 1 – i);

}

Return res;

}