



## Experiment Title: 2.1

**Student Name:** SANSKAR AGRAWAL

**UID:** 20BCS5914

**Branch:** CSE

**Section/Group:** MM\_806 / B

**Semester:** 5<sup>th</sup>

**D.O.P.:** 11-10-2022

**Subject Name:** PBLJ

**Subject Code:** 20CSP-321

### EXPERIMENT – 3.1

#### 1. Aim/Overview of the practical:

Create a palindrome creator application for making a longest possible palindrome out of given input string.

#### 2. Task to be done

Given a string, find the longest substring which is a palindrome.

#### 3. Code

```
import java.io.*; import
java.util.*;

public class Main {
    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        // String s = "hackerrekcahba";
        String s = sc.nextLine();
        char[] ca = s.toCharArray();
        StringBuilder sb = new StringBuilder();
        // Added sentinels to avoid bounds checking
        sb.append('!');
        for (int i = 0; i < ca.length; i++) {
            sb.append('#');
```

```
        sb.append(ca[i]);
    } sb.append('#');
    sb.append('@');

    int[] dp = new int[sb.length()]; int n = sb.length(); int C =
    0, R = 0; for (int i = 1; i < sb.length() - 1; i++) { int
    mirror_index = 2 * C - i;        dp[i] = (R > i) ? Math.min(R
    - i, dp[mirror_index]) : 0;

        while (sb.charAt(i + 1 + dp[i]) == sb.charAt(i - 1 - dp[i])) {
            dp[i]++;
        }

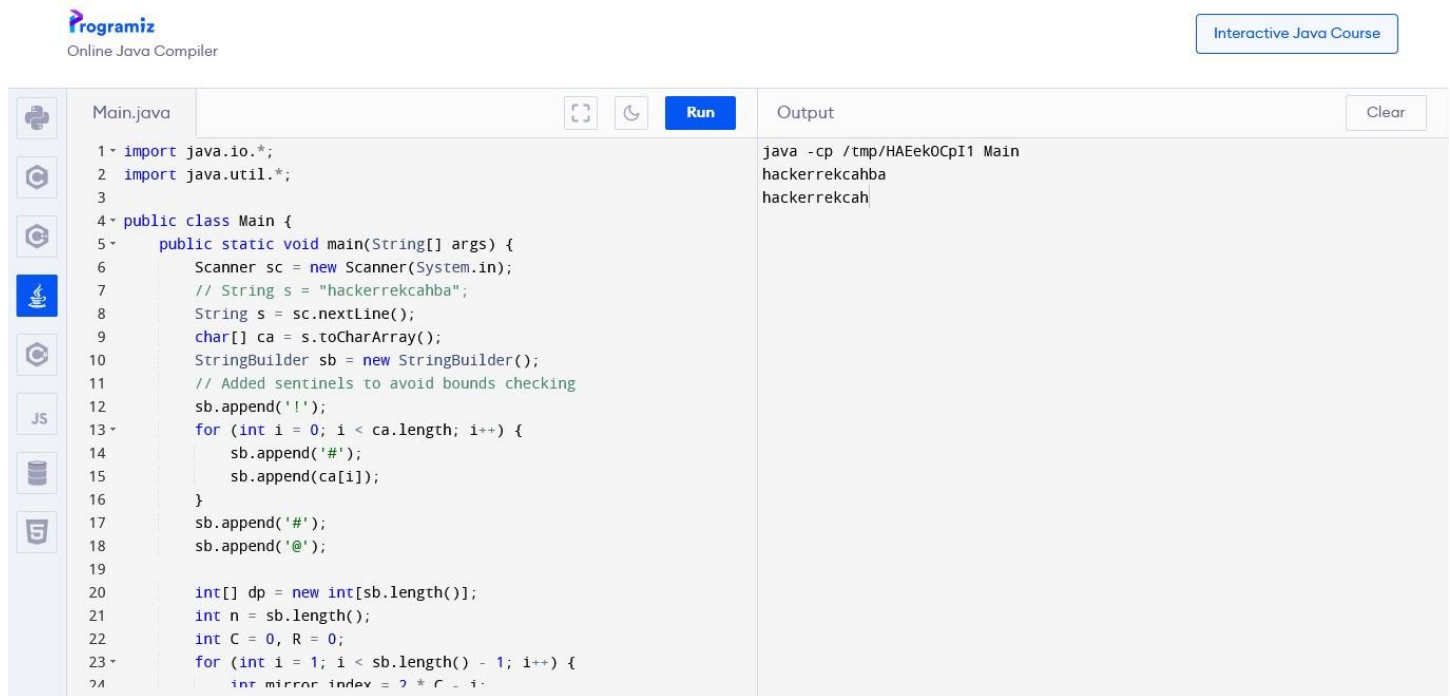
        if (i + dp[i] > R) {
            C = i;
            R = i + dp[i];
        }
    }

    int maxLen = 0; int centerIndex = 0;
    for (int i = 0; i < sb.length() - 1; i++) {
        if (dp[i] > maxLen) {
            maxLen = dp[i];
            centerIndex = i;
        }
    }

    System.out.println(s.substring((centerIndex - 1 - maxLen) / 2,
        (centerIndex - 1 + maxLen) / 2));

    }
```

## 4. Output



The screenshot shows the Programiz Online Java Compiler interface. On the left, there's a sidebar with icons for various programming languages. The main area displays a Java file named 'Main.java' with the following code:

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Main {
5     public static void main(String[] args) {
6         Scanner sc = new Scanner(System.in);
7         // String s = "hackerrekcahba";
8         String s = sc.nextLine();
9         char[] ca = s.toCharArray();
10        StringBuilder sb = new StringBuilder();
11        // Added sentinels to avoid bounds checking
12        sb.append('!');
13        for (int i = 0; i < ca.length; i++) {
14            sb.append('#');
15            sb.append(ca[i]);
16        }
17        sb.append('#');
18        sb.append('@');
19
20        int[] dp = new int[sb.length()];
21        int n = sb.length();
22        int C = 0, R = 0;
23        for (int i = 1; i < sb.length() - 1; i++) {
24            int mirror_index = 2 * C - i;
```

On the right, the 'Output' section shows the command and its result:

```
java -cp /tmp/HAEek0CpI1 Main
hackerrekcahba
hackerrekcah
```

## 5. Learning Outcome

string is called a palindrome string if the reverse of that string is the same as the original string. For example, radar, level, etc.

Similarly, a number that is equal to the reverse of that same number is called a palindrome number. For example, **3553**, **12321**, etc.