# Assignment 4 – Part 2: Palindromes

## Ryan Brinson

## 9/15/2023

## Output:

A computer screen with white and green text

Description automatically generated

## Code:

// Name: Ryan Brinson

// Class: CS 3305 W04

// Term: Fall 2023

//  Instructor:  Carla McManus

//  Assignment:  04-Part-2-Palindromes

import java.util.Scanner;

public class A4P2 {

    public static void main(String[] args) {

        Scanner input = new Scanner(System.in);

        // Get a word from the user and pass it to the constructor

        System.out.println("Enter word to test: ");

        Palindrom pali = new Palindrom(input.nextLine());

        if (pali.IsPalindrom() == true)

            System.out.println("\nThe word is a palindrome!");

        else

            System.out.println("\nThe word is not a palindrom.");

        input.close();

    }

}

class Palindrom{

    private char[] Stack\_1;

    private char[] Stack\_2;

    private char[] Stack\_3;

    private int top\_1;

    private int top\_2;

    private int top\_3;

    // Constructor that takes in word and fills the three stacks

    Palindrom(String word){

        // Initialize the tops of the stacks

        top\_1 = -1;

        top\_2 = -1;

        top\_3 = -1;

        // Convert the user word to a char array and

        // store in Stack 1

        Stack\_1 = word.toCharArray();

        // update the top of stack 1

        top\_1 = Stack\_1.length - 1;

        // Clone stack 1 into stack 2

        Stack\_2 = Stack\_1.clone();

        // Initialize the length of stack 3

        Stack\_3 = new char[Stack\_1.length];

        while (!is\_empty\_1()){

            push\_3(pop\_1());

        }

    }

    // Push values onto Stack 3

    private void push\_3(char c){

        //Check if the stack is full

        if(top\_3 == (Stack\_3.length - 1)) System.out.println("Stack 3 full");

        // If it is not full add the char to the top of the stack

        else {

            Stack\_3[top\_3 + 1] = c;

            top\_3++;

        }

    }

    // Test if Stack 1 is empty or not

    private boolean is\_empty\_1(){

        return top\_1 < 0;

    }

    // Pop the stack of stack 1

    private char pop\_1(){

        // set a temp value to the top value

        char temp = Stack\_1[top\_1];

        // set the top value to the null character

        Stack\_1[top\_1] = '\0';

        // decriment the stack

        top\_1--;

        // return the temp value

        return temp;

    }

    public boolean IsPalindrom(){

        // Convert Stack 2 and 3 to strings

        String temp1 = String.copyValueOf(Stack\_2);

        String temp2 = String.copyValueOf(Stack\_3);

        // Use the string .equals to compare if they are the same

        return temp1.equals(temp2);

    }

}