```java
import java.util.Scanner;

public class Question1_BMICalculation {

    public static String[][] calculateBMI(double[][] data) {
        String[][] result = new String[data.length][4];
        for (int i = 0; i < data.length; i++) {
            double weight = data[i][0];
            double heightCm = data[i][1];
            double heightM = heightCm / 100.0;
            double bmi = weight / (heightM * heightM);
            String status;
            if (bmi < 18.5) {
                status = "Underweight";
            } else if (bmi < 25) {
                status = "Normal weight";
            } else if (bmi < 30) {
                status = "Overweight";
            } else {
                status = "Obese";
            }
            result[i][0] = String.format("%.2f", heightCm);
            result[i][1] = String.format("%.2f", weight);
            result[i][2] = String.format("%.2f", bmi);
            result[i][3] = status;
        }
        return result;
    }

    public static void displayResults(String[][] data) {
        System.out.println("Height(cm)\tWeight(kg)\tBMI\tStatus");
        for (String[] row : data) {
            System.out.println(row[0] + "\t\t" + row[1] + "\t\t" + row[2] + "\t" + row[3]);
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
        double[][] data = new double[10][2];

        for (int i = 0; i < 10; i++) {
            System.out.print("Enter weight (kg) for person " + (i + 1) + ": ");
            data[i][0] = scanner.nextDouble();
            System.out.print("Enter height (cm) for person " + (i + 1) + ": ");
            data[i][1] = scanner.nextDouble();
        }

        String[][] results = calculateBMI(data);
        displayResults(results);

        scanner.close();
    }
}

import java.util.Scanner;

public class Question2_UniqueCharacters {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
            // Exception caught, return count
        }
        return count;
    }

    public static char[] findUniqueCharacters(String str) {
        int length = getStringLength(str);
        char[] unique = new char[length];
        int uniqueCount = 0;

        for (int i = 0; i < length; i++) {
            char current = str.charAt(i);
            boolean isUnique = true;
            for (int j = 0; j < i; j++) {
                if (str.charAt(j) == current) {
                    isUnique = false;
                    break;
                }
            }
            if (isUnique) {
                unique[uniqueCount++] = current;
            }
        }

        char[] result = new char[uniqueCount];
        System.arraycopy(unique, 0, result, 0, uniqueCount);
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        char[] uniqueChars = findUniqueCharacters(input);

        System.out.print("Unique characters: ");
        for (char c : uniqueChars) {
            System.out.print(c + " ");
        }
        System.out.println();

        scanner.close();
    }
}

import java.util.Scanner;

public class Question3_FirstNonRepeatingChar {

    public static char findFirstNonRepeatingChar(String str) {
        int[] freq = new int[256];
        int length = str.length();

        for (int i = 0; i < length; i++) {
            freq[str.charAt(i)]++;
        }

        for (int i = 0; i < length; i++) {
            if (freq[str.charAt(i)] == 1) {
                return str.charAt(i);
            }
        }
        return '\0'; // No non-repeating character found
    }
    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        char result = findFirstNonRepeatingChar(input);
        if (result != '\0') {
            System.out.println("First non-repeating character: " + result);
        } else {
            System.out.println("No non-repeating character found.");
        }

        scanner.close();
    }
}

import java.util.Scanner;

public class Question4_CharFrequency {

    public static String[][] findCharFrequency(String str) {
        int[] freq = new int[256];
        int length = str.length();

        for (int i = 0; i < length; i++) {
            freq[str.charAt(i)]++;
        }

        int uniqueCount = 0;
        for (int i = 0; i < 256; i++) {
            if (freq[i] > 0) {
                uniqueCount++;
            }
        }

        String[][] result = new String[uniqueCount][2];
        int index = 0;
        for (int i = 0; i < 256; i++) {
            if (freq[i] > 0) {
                result[index][0] = Character.toString((char) i);
                result[index][1] = Integer.toString(freq[i]);
                index++;
            }
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String[][] frequencies = findCharFrequency(input);

        System.out.println("Character\tFrequency");
        for (String[] pair : frequencies) {
            System.out.println(pair[0] + "\t\t" + pair[1]);
        }

        scanner.close();
    }
}

import java.util.Scanner;

public class Question5_FrequencyUsingUniqueChars {

    public static int getStringLength(String str) {
        int count = 0;
        try {
            while (true) {
                str.charAt(count);
                count++;
            }
        } catch (IndexOutOfBoundsException e) {
            // Exception caught, return count
        }
        return count;
    }

    public static char[] findUniqueCharacters(String str) {
        int length = getStringLength(str);
        char[] unique = new char[length];
        int uniqueCount = 0;
        for (int i = 0; i < length; i++) {
            char current = str.charAt(i);
            boolean isUnique = true;
            for (int j = 0; j < i; j++) {
                if (str.charAt(j) == current) {
                    isUnique = false;
                    break;
                }
            }
            if (isUnique) {
                unique[uniqueCount++] = current;
            }
        }

        char[] result = new char[uniqueCount];
        System.arraycopy(unique, 0, result, 0, uniqueCount);
        return result;
    }

    public static String[][] findFrequencyUsingUniqueChars(String str) {
        char[] uniqueChars = findUniqueCharacters(str);
        String[][] result = new String[uniqueChars.length][2];

        for (int i = 0; i < uniqueChars.length; i++) {
            int count = 0;
            for (int j = 0; j < str.length(); j++) {
                if (str.charAt(j) == uniqueChars[i]) {
                    count++;
                }
            }
            result[i][0] = Character.toString(uniqueChars[i]);
            result[i][1] = Integer.toString(count);
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String[][] frequencies = findFrequencyUsingUniqueChars(input);
        System.out.println("Character\tFrequency");
        for (String[] pair : frequencies) {
            System.out.println(pair[0] + "\t\t" + pair[1]);
        }

        scanner.close();
    }
}

import java.util.Scanner;

public class Question6_FrequencyUsingNestedLoops {

    public static String[][] findFrequency(String str) {
        char[] chars = str.toCharArray();
        int length = chars.length;
        int[] freq = new int[length];
        boolean[] visited = new boolean[length];

        for (int i = 0; i < length; i++) {
            if (visited[i]) continue;
            freq[i] = 1;
            for (int j = i + 1; j < length; j++) {
                if (chars[i] == chars[j]) {
                    freq[i]++;
                    visited[j] = true;
                }
            }
        }

        int uniqueCount = 0;
        for (int i = 0; i < length; i++) {
            if (!visited[i]) uniqueCount++;
        }

        String[][] result = new String[uniqueCount][2];
        int index = 0;
        for (int i = 0; i < length; i++) {
            if (!visited[i]) {
                result[index][0] = Character.toString(chars[i]);
                result[index][1] = Integer.toString(freq[i]);
                index++;
            }
        }
        return result;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a string: ");
        String input = scanner.nextLine();

        String[][] frequencies = findFrequency(input);

        System.out.println("Character\tFrequency");
        for (String[] pair : frequencies) {
            System.out.println(pair[0] + "\t\t" + pair[1]);
        }

        scanner.close();
    }
}

import java.util.Scanner;

public class Question7_PalindromeCheck {

    public static boolean isPalindromeLogic1(String str) {
        int start = 0;
        int end = str.length() - 1;
        while (start < end) {
            if (str.charAt(start) != str.charAt(end)) {
                return false;
            }
            start++;
            end--;
        }
        return true;
    }

    public static boolean isPalindromeLogic2(String str, int start, int end) {
        if (start >= end) {
            return true;
        }
        if (str.charAt(start) != str.charAt(end)) {
            return false;
        }
        return isPalindromeLogic2(str, start + 1, end - 1);
    }

    public static char[] reverseStringUsingCharAt(String str) {
        int length = str.length();
        char[] reversed = new char[length];
        for (int i = 0; i < length; i++) {
            reversed[i] = str.charAt(length - 1 - i);
        }
        return reversed;
    }

    public static boolean isPalindromeLogic3(String str) {
        char[] original = str.toCharArray();
        char[] reversed = reverseStringUsingCharAt(str);
        for (int i = 0; i < original.length; i++) {
            if (original[i] != reversed[i]) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter a text: ");
        String input = scanner.nextLine();

        System.out.println("Palindrome check using logic 1: " + isPalindromeLogic1(input));
        System.out.println("Palindrome check using logic 2: " + isPalindromeLogic2(input, 0,
input.length() - 1));
        System.out.println("Palindrome check using logic 3: " + isPalindromeLogic3(input));

        scanner.close();
    }
}

import java.util.Scanner;

public class Question8_AnagramCheck {

    public static boolean areAnagrams(String str1, String str2) {
        if (str1.length() != str2.length()) {
            return false;
        }

        int[] freq1 = new int[256];
        int[] freq2 = new int[256];

        for (int i = 0; i < str1.length(); i++) {
            freq1[str1.charAt(i)]++;
            freq2[str2.charAt(i)]++;
        }

        for (int i = 0; i < 256; i++) {
            if (freq1[i] != freq2[i]) {
                return false;
            }
        }
        return true;
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter first text: ");
        String text1 = scanner.nextLine();

        System.out.print("Enter second text: ");
        String text2 = scanner.nextLine();

        boolean result = areAnagrams(text1, text2);
        System.out.println("Are the two texts anagrams? " + result);

        scanner.close();
    }
}

import java.util.Scanner;

public class Question9_DisplayCalendar {

    public static String getMonthName(int month) {
        String[] months = {
            "January", "February", "March", "April", "May", "June",
            "July", "August", "September", "October", "November", "December"
        };
        if (month >= 1 && month <= 12) {
            return months[month - 1];
        }
        return "Invalid Month";
    }

    public static int getDaysInMonth(int month, int year) {
        int[] days = {
            31, 28, 31, 30, 31, 30,
            31, 31, 30, 31, 30, 31
        };
        if (month == 2 && isLeapYear(year)) {
            return 29;
        }
        if (month >= 1 && month <= 12) {
            return days[month - 1];
        }
        return 0;
    }

    public static boolean isLeapYear(int year) {
        return (year % 400 == 0) || (year % 4 == 0 && year % 100 != 0);
    }

    public static int getFirstDayOfMonth(int month, int year) {
        int y0 = year - (14 - month) / 12;
        int x = y0 + y0 / 4 - y0 / 100 + y0 / 400;
        int m0 = month + 12 * ((14 - month) / 12) - 2;
        int d0 = (1 + x + (31 * m0) / 12) % 7;
        return d0;
    }

    public static void displayCalendar(int month, int year) {
        System.out.println("    " + getMonthName(month) + " " + year);
        System.out.println("Su Mo Tu We Th Fr Sa");

        int firstDay = getFirstDayOfMonth(month, year);
        int daysInMonth = getDaysInMonth(month, year);

        for (int i = 0; i < firstDay; i++) {
            System.out.print("   ");
        }

        for (int day = 1; day <= daysInMonth; day++) {
            System.out.printf("%2d ", day);
            if ((day + firstDay) % 7 == 0) {
                System.out.println();
            }
        }
        System.out.println();
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        System.out.print("Enter month (1-12): ");
        int month = scanner.nextInt();

        System.out.print("Enter year: ");
        int year = scanner.nextInt();

        displayCalendar(month, year);

        scanner.close();
    }
}

import java.util.Random;
import java.util.Scanner;

public class Question10_DeckOfCards {

    public static String[] initializeDeck() {
        String[] suits = {"Hearts", "Diamonds", "Clubs", "Spades"};
        String[] ranks = {"2", "3", "4", "5", "6", "7", "8", "9", "10", "Jack", "Queen", "King", "Ace"};
        String[] deck = new String[suits.length * ranks.length];
        int index = 0;
        for (String suit : suits) {
            for (String rank : ranks) {
                deck[index++] = rank + " of " + suit;
            }
        }
        return deck;
    }

    public static void shuffleDeck(String[] deck) {
        Random random = new Random();
        int n = deck.length;
        for (int i = 0; i < n; i++) {
            int randomCardNumber = i + random.nextInt(n - i);
            String temp = deck[i];
            deck[i] = deck[randomCardNumber];
            deck[randomCardNumber] = temp;
        }
    }

    public static String[][] distributeCards(String[] deck, int numCards, int numPlayers) {
        if (numCards > deck.length || numCards < numPlayers) {
            return null; // Cannot distribute
        }
        String[][] players = new String[numPlayers][numCards / numPlayers];
        int cardIndex = 0;
        for (int i = 0; i < numCards / numPlayers; i++) {
            for (int j = 0; j < numPlayers; j++) {
                players[j][i] = deck[cardIndex++];
            }
        }
        return players;
    }

    public static void printPlayersCards(String[][] players) {
        for (int i = 0; i < players.length; i++) {
            System.out.println("Player " + (i + 1) + "'s cards:");
            for (String card : players[i]) {
                System.out.println(card);
            }
            System.out.println();
        }
    }

    public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);

        String[] deck = initializeDeck();
        shuffleDeck(deck);

        System.out.print("Enter number of cards to distribute: ");
        int numCards = scanner.nextInt();

        System.out.print("Enter number of players: ");
        int numPlayers = scanner.nextInt();
        String[][] players = distributeCards(deck, numCards, numPlayers);
        if (players == null) {
            System.out.println("Cannot distribute " + numCards + " cards to " + numPlayers + "
players.");
        } else {
            printPlayersCards(players);
        }

        scanner.close();
    }
}
```