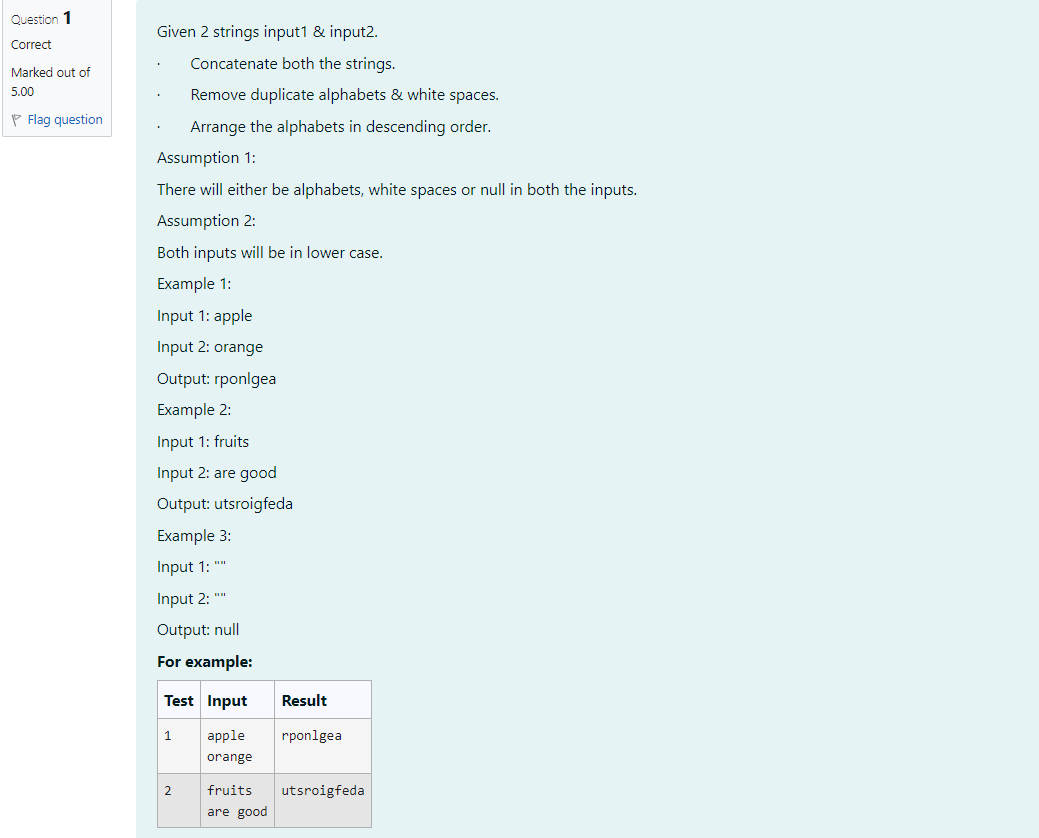
OBJECT ORIENETED PROGRAMMING USING JAVA

NAME : P.R.DHANVIN

DEPT & SEC : CSE & B

ROLL NO: 230701071

WEEK : 6



import java.util.\*;

public class String Processor {

public static String processStrings (String input1, String input2) {

String concatenated input1 input2;

if (concatenated.trim().isEmpty()) { return "null";

}

Set<Character> charSet = new LinkedHashSet<>();

for (char c: concatenated.toCharArray()) {

if (c != ') {

charSet.add(c);

}

}

List<Character> charList = new ArrayList<>(charSet); charList.sort(Collections.reverseOrder());

StringBuilder output new StringBuilder();

for (char c charList) {

output.append(c);

}

return output.toString();

}

public static void main(String[] args) {

Scanner scanner = new Scanner(System.in);

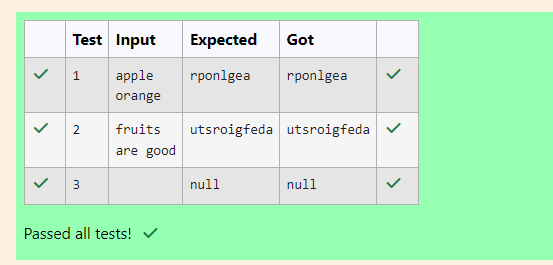
String input1 = scanner.nextLine();

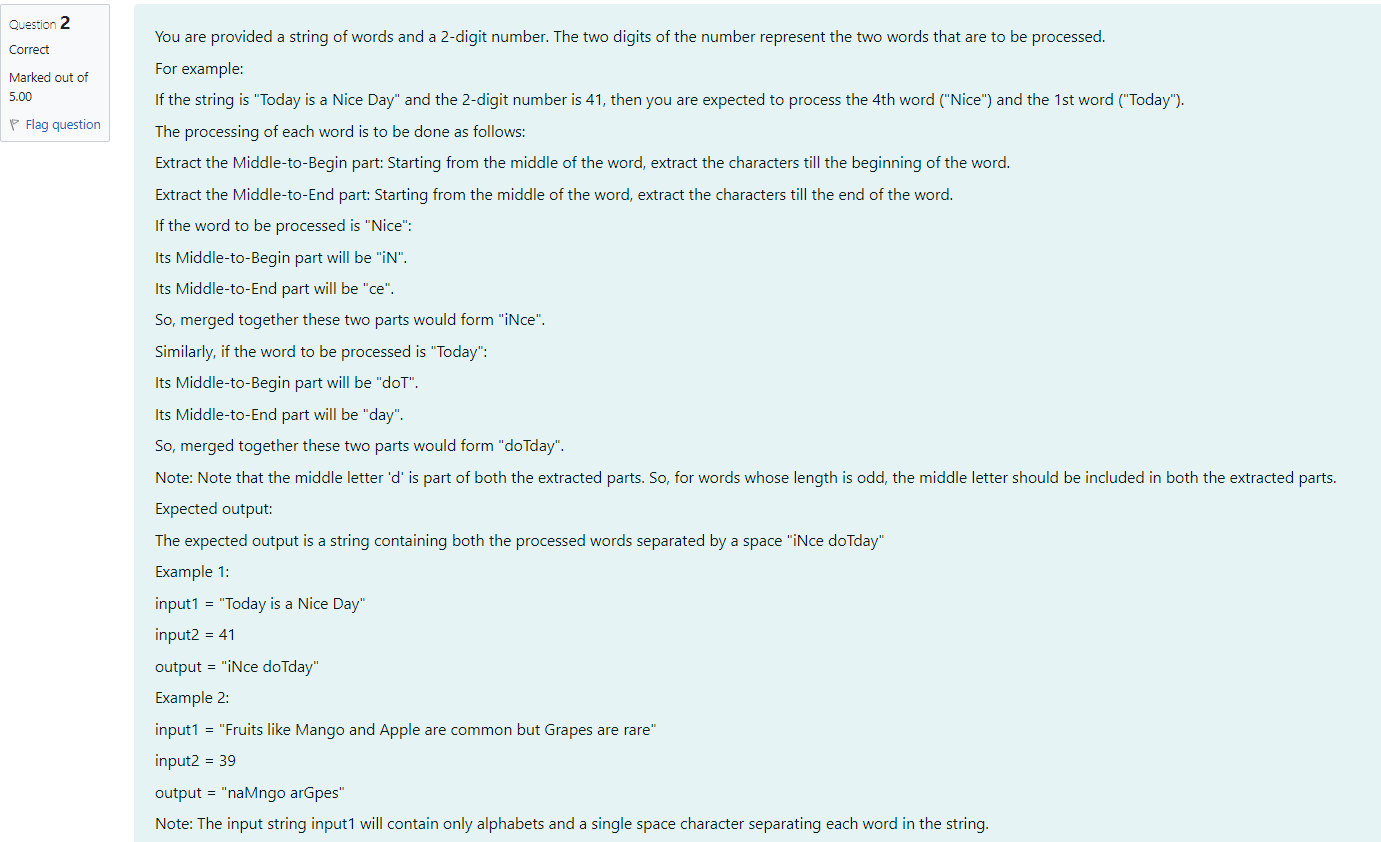
String input2 scanner.nextLine();

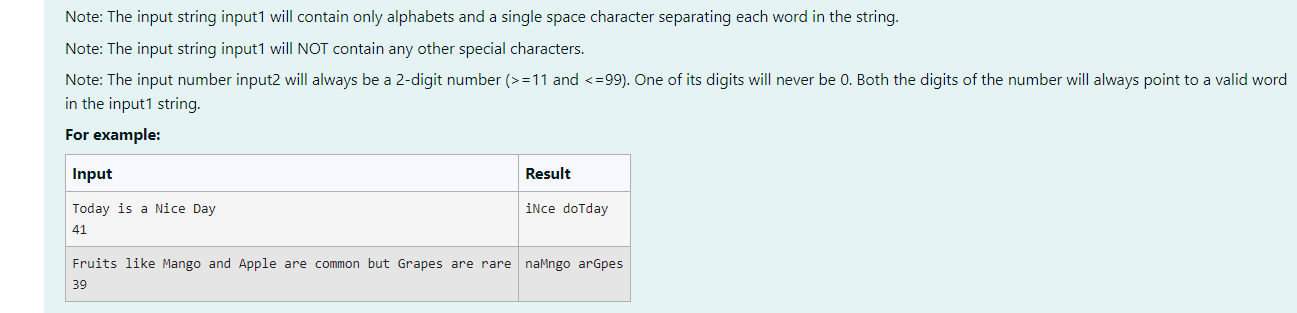
String result = processStrings (input1, input2); System.out.println(result);

}

}







import java.util.Scanner;

public class WordProcessor {

public static String processWord(String word) {

int length word.length();

int middleIndex length / 2; // Get the middle index

String middleToBegin;

String middleToEnd;

if (length % 2 == 0) {

middleToBegin = new StringBuilder(word.substring(0, middle Index)).reverse().toString(); middleToEnd word.substring(middleIndex);

} else {

middleToBegin = new StringBuilder(word.substring(0, middle Index + 1)).reverse().toString();

middleToEnd = word.substring(middleIndex);

}

return middleToBegin middleToEnd;

}

public static String processInput(String input1, int input2) {

String[] words input1.split(" ");

firstWordIndex Character.getNumericValue (Integer.toString(input2).charAt(0)) 1; // Convert to 8-index

1; // Convert to

0-index

int int secondWordIndex Character.getNumericValue (Integer.toString(input2).charAt(1))

String firstProcessedklord String secondProcessedWord processWord (words [firstWordIndex]); processWord (words [secondWordIndex]);

return firstProcessedWord+"" secondProcessedWord;

}

public static void main(String[] args) {

Scanner scanner new Scanner(System.in);

String input1 scanner.nextLine();

int input2

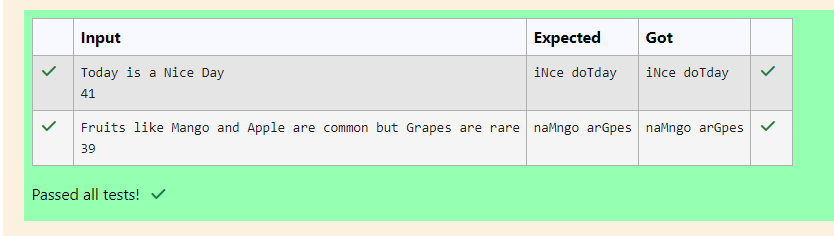
scanner.nextInt();

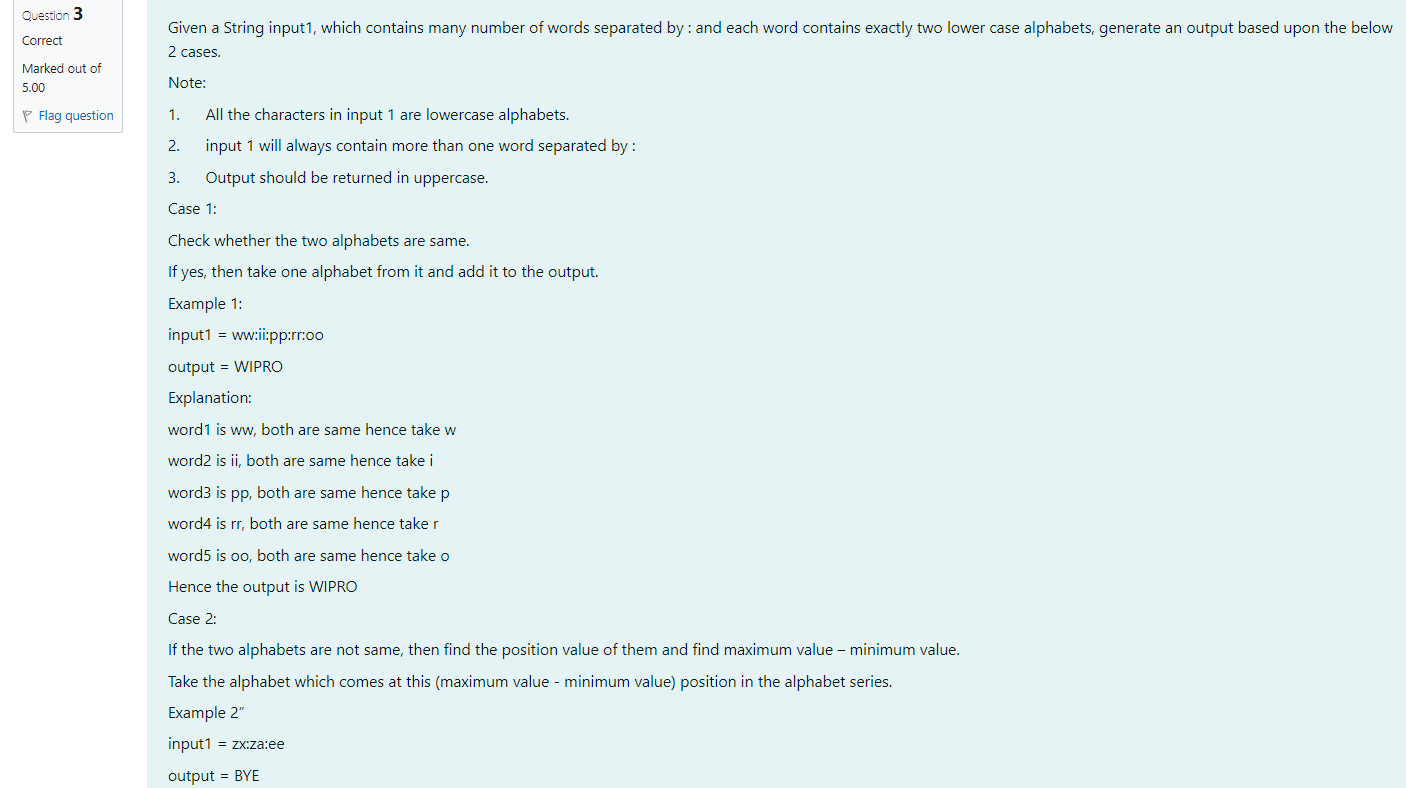
String result processInput (input1, input2);

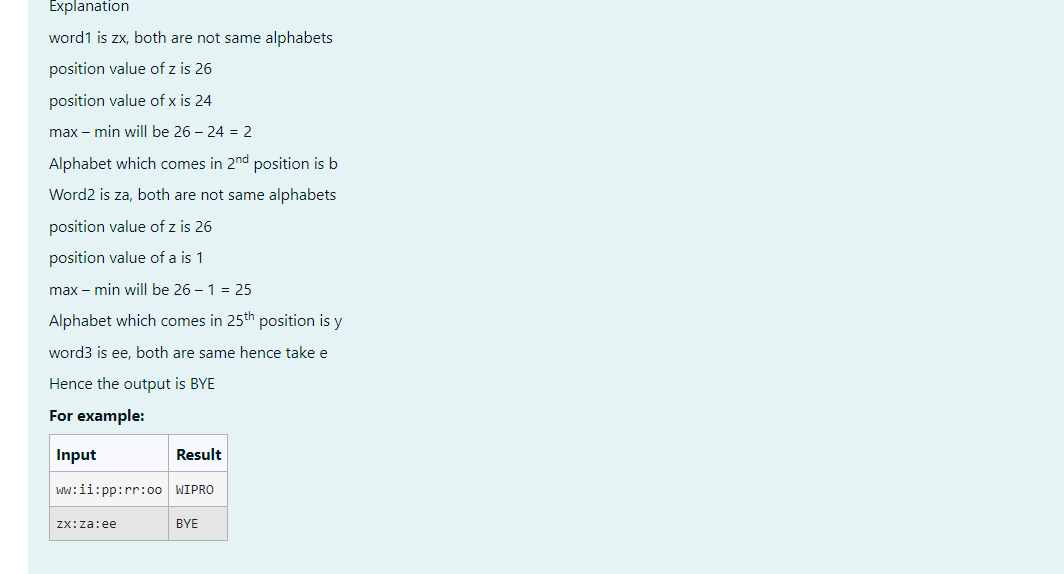
System.out.println(result);

}

}







import java.util.Scanner;

public class WordProcessor {

public static String processInput(String input) { StringBuilder output = new StringBuilder(); String[] words input.split(":");

for

(String word words) { char firstChar word.charAt(0);

char secondChar word.charAt(1);

if (firstChar== secondChar) {

output.append(Character.toUpperCase(firstChar));

} else {

int firstPos firstChar 'a' + 1; // Position in alphabet (1-26)

int secondPos secondChar 'a' + 1;

int maxPos = Math.max(firstPos, secondPos);

int minPos = Math.min(firstPos, secondPos);

int positionValue = maxPos minPos;

char resultChar = (char) ('a' positionValue 1); output.append(Character.toUpperCase (resultChar));

}

}

return output.toString();

}

public static void main(String[] args) {

Scanner scanner new Scanner(System.in); String input scanner.nextLine();

if (input.isEmpty() || input.contains(":")) {

System.out.println("Invalid input. Please ensure it contains multiple words separated by:".");

return;

}

String result processInput(input);

System.out.println(result);

}

}

