На входе два массива (карта и группировка танков), каждый описывается одинаково: количество строк, количество столбцов и набор строк, разделённых пробелом. Каждый символ строки описывает один элемент карты.

Например, дана карта:

1234

2345

0987

Этот массив размером 3\*4 будет представлен на вводе так:

3 4 1234 2345 0987

Второй массив представляется аналогично первому.

**Задача**: проверить, входит ли второй массив в первый (в двумерном виде).

Например, второй массив может быть таким:

34

98

или

2 2 34 98

Он входит в состав первого массива.

Функция

bool TankRush(int H1, int W1, string S1, int H2, intW2, string S2)

первыми тремя параметрами получает исходную карту, а следующими тремя -- карту, которая ищется в исходной.

TankRush возвращает true, если вторая карта содержится в первой.

[как постить решение](https://skillsmart.ru/algo/lvl1/how.html)

предварительная версия:)))

1. заполняем два строковых массива из данных Srting (Arr1, Arr2)
2. выполняем проверку на совпадение первой строки второго массива

со всеми строками первого массива (в порядке очереди)

т.е. смотрим входит ли 34 в какую либо строку первого массива, и если нет – continue

3. при совпадении (нахождении) искомой подстроки в строке отмечаем ее позицию (примерно так)

matchIndex = Arr[j].indexOf(Arr1[safr]);

и дальше идем на следующую строку первого массива и ищем совпадение по конкретному индексу

решение ниже работает ТОЛЬКО ЕСЛИ НАЙДЕНО ОДНО СОВПАДЕНИЕ В ОДНОЙ СТРОКЕ

т.е. «1234 2398 0987» и «34 98» выдаст true

а «3434 2398 0987» и «34 98» выдаст false

поэтому надо написать реализацию, когда при проверке столбцов и невыполнении условия – продолжится проверка первой строки с места второго совпадения

Тест закончился ошибкой:  
Исключение в TankRush

(4,6,'029402 560202 029694 780288',2,2,'02 94')

Метод нахождения нескольких подстрок в строке (повторяющихся)

С выводом массива индексов

public static int[] findElementIndex(String inputArr, String element) {  
  
 int offset = 0;  
 ArrayList<Integer> myNumbers = new ArrayList<>();  
 StringBuilder sb1 = new StringBuilder(inputArr);  
  
 for (int j = 0; j < sb1.length(); j++) {  
  
 if (sb1.indexOf(element) == sb1.lastIndexOf(element)){  
 myNumbers.add(sb1.indexOf(element) + offset);  
 break;  
 }  
 else {  
 myNumbers.add(sb1.indexOf(element) + offset);  
 offset = offset + sb1.indexOf(element) + element.length();  
 sb1.delete(0, sb1.indexOf(element) + element.length());  
 }  
 }  
  
 int [] result = new int[myNumbers.size()];  
 for (int k = 0; k < result.length; k++) {  
 result[k] = myNumbers.get(k);  
 }  
 return result;  
 }

далее вставляем этот метод в наш основной класс

решение ниже еще не исправлено (старый вариант):

public class Tanks {  
  
 public static boolean TankRush(int H1, int W1, String S1, int H2, int W2, String S2){  
 boolean found = false;  
 int arr2row1 = 0; //second array first row  
 int start1 = 0;  
 int start2 = 0;  
 String Arr1[] = new String[H1];  
 String Arr2[] = new String[H2];  
 int matchIndex = -1;  
  
 //filling up both arrays  
 for (int i = 0; i < H1; i++){  
 Arr1[i] = (S1.substring(start1, start1 + W1));  
 start1 = start1 + W1 + 1;  
 System.*out*.println(Arr1[i]);  
 }  
 for (int i = 0; i < H2; i++){  
 Arr2[i] = (S2.substring(start2, start2 + W2));  
 start2 = start2 + W2 + 1;  
 System.*out*.println(Arr2[i]);  
 }  
  
 //main loop  
 for (int j = 0; j < H1; j++){  
 System.*out*.println("checking row " + j);  
 if (!(Arr1[j].contains(Arr2[arr2row1]))){  
 continue;  
 }  
 if (j == H1 - 1){  
 System.*out*.println("last row match, exiting");  
 break;  
 }  
  
 matchIndex = Arr1[j].indexOf(Arr2[arr2row1]);  
 System.*out*.println("initial match at row: " + j + ", matchIndex: " + matchIndex + ", checking cols");  
  
 for (int r = j + 1, s = 0; r < H2; r++, s++){  
 if (!((Arr1[r].contains(Arr2[s]) && Arr1[r].indexOf(Arr2[s]) == matchIndex))) {  
 System.*out*.println("no vertical match, returning to row " + j);  
  
 break;  
 }  
 found = true;  
 }  
 }  
 return found;  
 }

ниже правильное решение, высчитывает все что задано в задаче

(с комментариями)

import java.util.\*;  
  
public class Tanks {  
  
 public static int[] findElementIndex(String inputArr, String element) {  
  
 int offset = 0;  
 ArrayList<Integer> myNumbers = new ArrayList<>();  
 StringBuilder sb1 = new StringBuilder(inputArr);  
  
 for (int j = 0; j < sb1.length(); j++) {  
 if (sb1.indexOf(element) == sb1.lastIndexOf(element)){  
 myNumbers.add(sb1.indexOf(element) + offset);  
 break;  
 }  
 else {  
 myNumbers.add(sb1.indexOf(element) + offset);  
 offset = offset + sb1.indexOf(element) + element.length();  
 sb1.delete(0, sb1.indexOf(element) + element.length());  
 }  
 }  
  
 int [] result = new int[myNumbers.size()];  
 for (int k = 0; k < result.length; k++) {  
 result[k] = myNumbers.get(k);  
 }  
 return result;  
 }  
  
 public static boolean TankRush(int H1, int W1, String S1, int H2, int W2, String S2) {  
 boolean found = false;  
 int arr2row1 = 0; //second array first row  
 int start1 = 0;  
 int start2 = 0;  
 String Arr1[] = new String[H1];  
 String Arr2[] = new String[H2];  
  
  
 //filling up both arrays  
 for (int i = 0; i < H1; i++) {  
 Arr1[i] = (S1.substring(start1, start1 + W1));  
 start1 = start1 + W1 + 1;  
 System.*out*.println(Arr1[i]);  
 }  
 for (int i = 0; i < H2; i++) {  
 Arr2[i] = (S2.substring(start2, start2 + W2));  
 start2 = start2 + W2 + 1;  
 System.*out*.println(Arr2[i]);  
 }  
  
 //main loop  
 for (int j = 0; j < H1; j++) {  
 System.*out*.println("checking row " + j);  
 if (!(Arr1[j].contains(Arr2[arr2row1]))) {  
 continue;  
 }  
 if (j == H1 - 1) {  
 System.*out*.println("last row match, exiting");  
 break;  
 }  
  
 int[] matchIndex = *findElementIndex*(Arr1[j], Arr2[arr2row1]);  
 int strLength = matchIndex.length;  
 System.*out*.println("match(es) at row: " + j + ", matchIndex: " + (Arrays.*toString*(matchIndex)));  
  
 //checking all matches in one row (j)  
 for (int m = 0; m < strLength; m++) {  
  
 for (int r = j + 1, s = 1; s < H2; r++, s++) {  
  
 System.*out*.println("checking Arr1 row " + r + " at index " + matchIndex[m] + " vs Arr2 row " + s) ;  
  
 if (!((Arr1[r].contains(Arr2[s]) && Arr1[r].indexOf(Arr2[s]) == matchIndex[m]))) {  
 System.*out*.println("no vertical match, returning to row " + j);  
  
 break;  
 }  
 System.*out*.println("match found!!! at Arr1 row: " + r + ", index: " + matchIndex[m]);  
 found = true;  
 }  
 if (found) break;  
  
 }  
 if (found) break;  
  
 }  
 return found;  
 }  
  
  
  
  
 public static void main(String[] args) {  
  
 String Str1 = "029402 560202 029694 780288";  
 String Str2 = "02 94";  
  
 System.*out*.println(*TankRush*(4, 6, Str1, 2, 2, Str2));  
  
  
 }  
}