

P1: Vector and ArrayDeque

本題須使用 Vector 以及 ArrayDeque，有關其用法請自行查詢 Java API Library。

*****本題須依照題目題意作答，不然不予以給分*****

Problem

題目內容為給你一組數字，以 Vector<Integer> vector 儲存，先升序 sort 之後，依序以特定方式儲存到 ArrayDeque<Integer> array，方式為假設 vector 的數字為偶數，則加到 array 的尾端，假設 vector 的數字為奇數，則加到 array 的前端，且在每組測資結束後以 removeFirst() 的方式清空 ArrayDeque。

Input

輸入有多筆測資，每行代表一組測資，每組測資會輸入一段數字 (數字的大小為 >0 ， $<2,147,483,647$)，數字的數目不一定，輸入 0 為一組測資的結束。

Output

依照 ArrayDeque array 儲存的順序依序輸出，輸出的值之間請以一個空格分開。

Sample Input

```
10 1 4 8 5 2 9 7 6 3 0↵
100 200 300 25 69 85 71 63 22 62 585 25
63 15 15 36 14 36 85 92 62 22 2 1 0↵
```

Sample Output

```
9 7 5 3 1 2 4 6 8 10↵
585 85 85 71 69 63 63 25 25 15 15 1
2 14 22 22 36 36 62 62 92 100 200
300↵
```

P2: The Complex Class

Problem

A complex number is a number in the form $a + bi$, where a and b are real numbers and i is $\sqrt{-1}$. The numbers a and b are known as the real part and imaginary part of the complex number, respectively. You can perform addition, subtraction, multiplication, and division for complex numbers using the following formula:

$$a + bi + c + di = (a+c) + (b+d)i$$

$$a + bi - (c + di) = (a-c) + (b-d)i$$

$$(a + bi) * (c + di) = (ac-bd) + (bc+ad)i$$

$$(a + bi) / (c + di) = (ac+bd)/(c^2+d^2) + (bc-ad)i/(c^2+d^2)$$

You can also obtain the absolute value for a complex number using the following formula:

$$|a + bi| = \sqrt{a^2 + b^2}$$

Design a class named **Complex** for representing complex numbers and the methods **add**, **subtract**, **multiply**, **divide**, and **abs** for performing complex number operations, and override **toString** method for returning a string representation for a complex number. The **toString** method returns $(a + bi)$ as a string. If b is 0, it simply returns a . Your **Complex** class should also implement the **Cloneable** interface.

Provide two constructors **Complex()** and **Complex(a, b)**. **Complex()** creates a Complex object for number 0 and **Complex(a, b)** creates a Complex object with $a + bi$.

Input

There are many input lines. Each line is a test case and it has four double values a , b , c , and d corresponding two complex numbers $a + bi$ and $c + di$.

Output

For each input case, please reference the following sample output case. The output of each test case are separated new lines

Sample Input

Sample Output

3.5 5.5 -3.5 1↵	$(3.5 + 5.5i) + (-3.5 + 1.0i) = 6.5i$
1 2 1 -2↵	$(3.5 + 5.5i) - (-3.5 + 1.0i) = 7.0 + 4.5i$
	$(3.5 + 5.5i) * (-3.5 + 1.0i) = -17.75 + -15.75i$
	$(3.5 + 5.5i) / (-3.5 + 1.0i) = -0.5094 + -1.7i$
	$ 3.5 + 5.5i = 6.519202405202649$
	↵
	$(1 + 2i) + (1 - 2i) = 2$
	$(1 + 2i) - (1 - 2i) = 4i$
	$(1 + 2i) * (1 - 2i) = 5$
	$(1 + 2i) / (1 - 2i) = -0.6 + 0.8i$
	$ 1 + 2i = 2.2360679775499790$

P3: 座標點陣列排序

***** 未依照題目規定不予計分 *****

Problem

請定義一個整數座標點(Point)類別，此 Point 類別可實作 Comparable 介面，兩點 P1 與 P2 之比較原則如下：

- $P1 < P2$ if P1 到原點(0,0)的距離 (distance) $<$ P2 到原點的距離
- 如 P1 與 P2 到原點的距離相等
 - ✧ $P1 < P2$ if P1 之 X 座標 $<$ P2 之 X 座標
 - ✧ 如 P1 與 P2 之 X 座標相等， $P1 < P2$ if P1 之 Y 座標 $<$ P2 之 Y 座標

Hint: 如 Point 無實作 Comparable 介面，可實作 Comparator 物件，排序時可以使用。

Input

本題會有多筆測資，每筆測資先輸入數字 N，代表有 N 個座標點，接著輸入 N 行資料，每行有兩整數分別代表 X 座標與 Y 座標。

Output

輸出上述(x, y)由小到大的排序，以及 x+y 相加最大的值與點位 ex: max num: 19518 point(9829,9689)，每組輸出以空白行隔開。(詳細請見 Sample Output)

Sample Input

Sample Output

3↵	(1,3)↵
1 3↵	(2,6)↵
6 2↵	(6,2)↵
2 6↵	max num: 8↵
6↵	point: (6,2)↵
5 6↵	↵
1 2↵	(1,2)↵
4 3↵	(2,3)↵
2 3↵	(3,4)↵
6 5↵	(4,3)↵
3 4↵	(5,6)↵
	(6,5)↵
	max num: 11↵
	point: (6,5)↵

P4: 四則運算式

Problem

要求你輸入一個運算式字串 S ，請先將字串 S 分割成多個 tokens，先將所有運算元 (operand) tokens 依序輸出，再輸出運算符號(operator) token，最後輸出計算結果，運算符號有 $+$ 、 $-$ 、 $*$ 及 $/$ 。

Input

本題輸入有多筆測資，每筆測資會輸入一個運算式字串，token 之間以空白字元分開，每個運算式會有 3 個運算元及 2 個運算符號。

Output

先輸出運算式中之運算元 (operand) token，再輸出運算符號(operator) token，輸出 token 之間以空白字元分開，最後輸出計算結果，**小數點以下印兩位**，正數請印出正號($+$)。每筆測資輸出間，以空白行分開。

Sample Input

Sample Output

3 + 2 / 8	3 2 8 + /
2 * 5 + 20	+3.25
2 - 20 / 5	
	2 5 20 * +
	+30.00
	2 20 5 - /
	-2.00

P5: Bin Packing Using First Fit

Problem

The bin packing problem is to pack the objects of various weights into containers. Assume each container can hold a maximum of **n** pounds. The program uses an algorithm that places an object into first bin in which it would fit. Your program should prompt the user to enter the number **n**, the number of objects **m**, and the weight of each object. The program display the total number of containers needed to pack the objects and the contents of each container. Here is a sample output of a test case inputting **n** = 10, **m** = 6, and the weights of all objects 7 5 2 3 5 8. The output is shown below:

```
4
Container 1 contains objects with weight 7 2
Container 2 contains objects with weight 5 3
Container 3 contains objects with weight 5
Container 4 contains objects with weight 8
```

Input

Each test case has two input lines. The first line contains two integer **n** and **m**. The second line contains the weights of all objects. There are many test cases.

Output

For each test cases, the first line outputs the number (**k**) of bins needed to pack all objects followed by **k** lines of output in which each line displays the content of each container. There is a new line between two test cases.

Sample Input

Sample Output

10 6↵	4↵
7 5 2 3 5 8↵	Container 1 contains objects with weight 7 2↵
12 10↵	Container 2 contains objects with weight 5 3↵
5 8 7 10 6 9 5 3 2 1↵	Container 3 contains objects with weight 5↵
	Container 4 contains objects with weight 8↵
	↵
	5↵
	Container 1 contains objects with weight 5 7↵
	Container 2 contains objects with weight 8 3 1↵
	Container 3 contains objects with weight 10 2↵
	Container 4 contains objects with weight 6 5↵
	Container 5 contains objects with weight 9↵