ISSS 622
Python Programming and Data Analysis

Assignment 1

[Question 1.31] Palindromic Clock Time (5 marks)

A digital clock which displays time in the format of MM:DD:hh:mm (Month-Month, Day-Day, hour-hour in 24 hours format, and minute-minute) will occasionally show a string that looks the same backward as forward, which we call a palindromic clock time. For example, if it is 1220hrs on the day of Feb 21st 2021, the displayed clock time is 02:21:12:20. In this question, your task is to list all palindromic clock time in chronological order for a given interval defined by two calendar days.

Input & Output

The first line of the input indicates the number of test cases below. Each test case is described by one line containing two specific dates separated by a space, and the dates are written in the format of "dd/mm/yyyy". It is guaranteed (meaning there is no need to write code to check) that (i) the dates are valid, (2) the first date is earlier than the second date and (3) the duration between the two dates is not longer than 1 year. For each test case, print all palindromic clock time between the two days inclusive (meaning from the 0000hrs of the first date to 2359hrs of the second date) in the format of MM:DD:hh:mm where hh is in 24 hours format. Palindromic clock time should be arranged in chronological order, one per line. Print one extra line of 11 dashes (to align with other output) in between two test cases. No dash lines shall be printed at the beginning or at the end.

Sample Input

```
3
21/02/2021 21/02/2021
01/03/2021 20/04/2021
10/12/2021 07/01/2022
```

Sample Output

```
02:21:12:20
-----------
03:01:10:30
03:02:20:30
03:10:01:30
03:11:11:30
```

```
03:12:21:30
03:20:02:30
03:21:12:30
03:22:22:30
03:30:03:30
03:31:13:30
04:01:10:40
04:02:20:40
04:10:01:40
04:11:11:40
04:12:21:40
04:20:02:40
-----------
12:10:01:21
12:11:11:21
12:12:21:21
12:20:02:21
12:21:12:21
12:22:22:21
12:30:03:21
12:31:13:21
01:01:10:10
01:02:20:10
```

[Question 1.32] Social Distancing in Vaccination Centers (10 marks)

In vaccination centers, people are required to sit down at a hall for half an hour after taking the vaccine. The social distance of 1 meter should be observed to ensure Covid-19 safety measures. However, we would like to place as many chairs as possible in the hall to handle largest possible crowd. Given the length and width of a rectangular hall, your task is to compute the maximum number of chairs to be placed with pairwise distance at least 1 meter apart. To simplify the problem, you may ignore the size of the chairs, and consider them as points which are at least 1 meter apart.

Input & Output

The first line of the input indicates the number of test cases below. Each test case is described by two integers separated by a space, they are the length and width of a rectangular hall. Both length and width are integers between 5 and 100 inclusive.

Sample Input

```
2
8 9
10 10
```

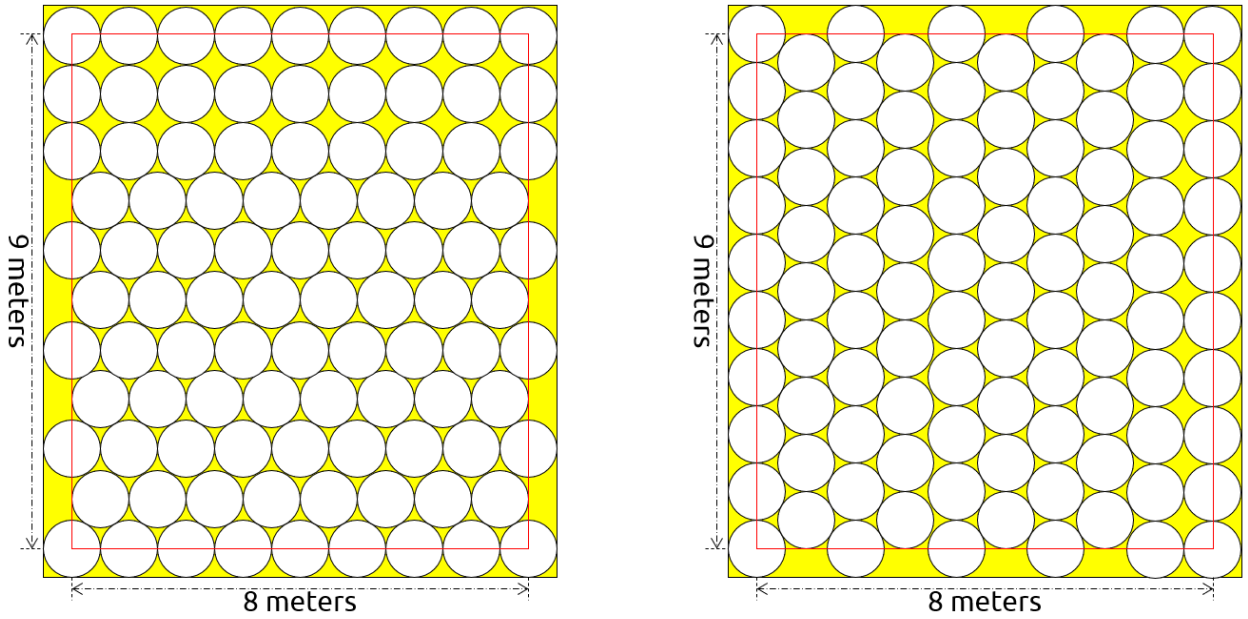
Sample Output

```
96
128
```

Hint: Think about packing cans of soda in a box, or packing circles in a rectangle: how many circles can you pack in a rectangle? If (i) each chair is modeled as a circle of radius 0.5 meters,

and (ii) the rectangle is enlarged by 0.5 meters on all four sides, the maximum number of chairs you can place equals to the number of circles you can pack in the enlarged rectangle.

There are two ways to pack circles in a rectangle, rectangular pattern and triangular pattern: www.engineeringtoolbox.com/circles-within-rectangle-d_1905.html. Circles are more squeezed in triangular pattern because it allows the next row of circles to "squeeze in the gap" of the two neighboring circles in the previous row. However, you have one circle fewer in alternate rows. Rectangular pattern does not make full use of the gaps, but it ensures maximum of circles in each row. But if we combine these two patterns together, we may pack more than either pattern. For example, for a hall of 8 meters long and 9 meters wide, at most 90 chairs or 95 chairs can be placed using rectangular pattern or triangular pattern only. However, if we consider mixing the two patterns, we may pack more circles. Using the same example, we can pack 95 circles if we mix rectangular pattern and triangular pattern along the width (as shown in left figure (a), if only triangular pattern was used, there would only be 94 circles), and we can pack 96 circles if we mix the two patterns along the length (as shown in right figure (b), if only triangular pattern was used, there would only be 95 circles). You need to think how to mix the two patterns along either dimension to pack as many as possible.



(a) 95 circles

(b) 96 circles

[Question 1.33] Flash Flooded Skyline (10 marks)

Flash flood is a serious problem for modern cities. A team of IT professionals are engaged to describe how severe flash floods can be, and you are assigned to the core task which computes the height of the flash floods at different parts of a city in a simulated setting with a specified rainfall. The following assumptions help you simply the problem.

- Buildings are lined up in a single file, and all building are of 20 meters wide, but their height can be different. Buildings are next to one another without any gap.
- When it rains, rainwater falling on one tall building will flow to the shorter buildings if there are. If the building on the left side is shorter but the building on the right side is taller, water will flow to the left side, and vice versa. If both buildings on the left side and on the right side are shorter, half of the water flow to the left side and the remaining half flow to the right side.
- There are imaginary walls on both left end and right end, which are taller than all buildings.
- When the space on top of a shorter building is filled with water and it levels up with a taller building beside, water will start accumulate in the space supported by both buildings.
- A piece of empty land is considered as a building of height 0, so rainwater from the buildings at the two sides will start accumulating in this space first.
- You are given the skyline of the city, i.e., the heights of all buildings, and the list of possible rainfalls (in meter, these values look ridiculous as we are doing a simulation from a 3D world into a 2D display).
- You are expected to compute how the skyline of the city looked like with each specified rainfall. A skyline is described by a list of horizontal line segments with height and length.

Input

You are given two lists, a list of building heights from left to right, and a list of possible rainfalls. The heights are integers but a value in the rainfall list can be a decimal. The maximum length of the two lists is 50 (the list of building heights) and 10 (the list of rainfalls) respectively.

Sample Input

```
4 18 5 0 10 10 60 40 20 30 28 36
20 10 30 50
```

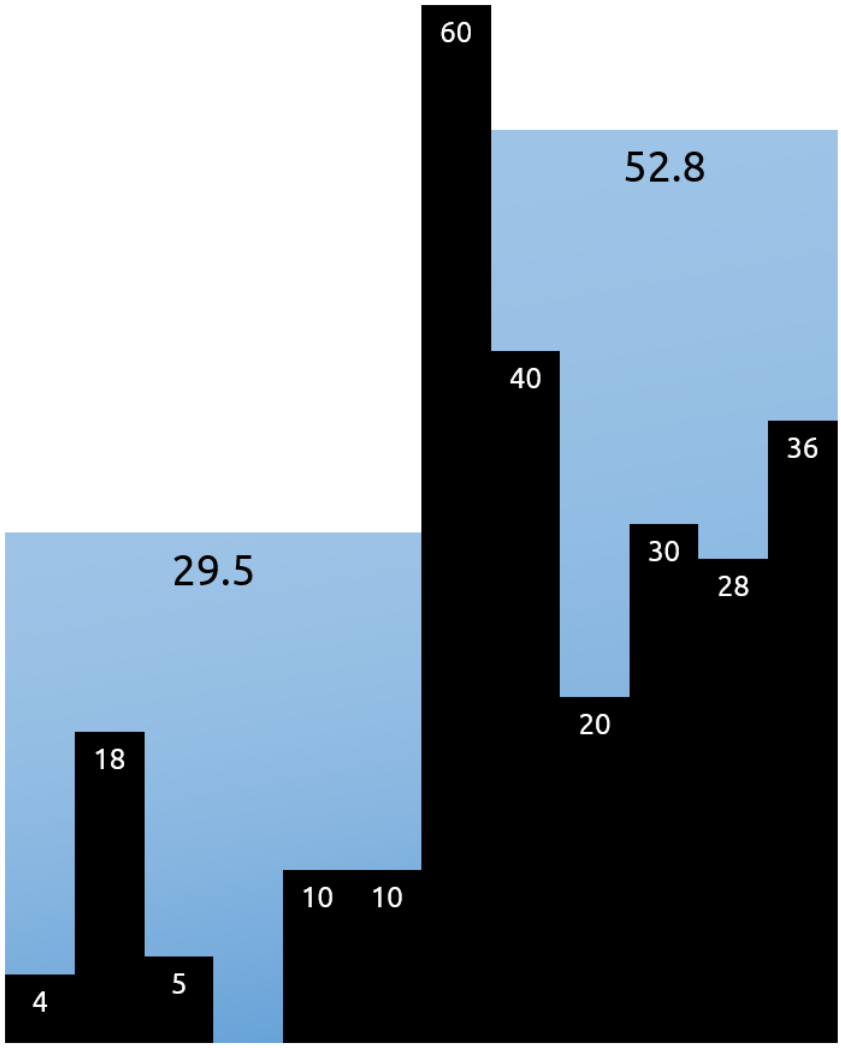
Output

For each specified rainfall value, you need to describe the skyline of the city with flood in one line. A skyline is a list of horizontal line segments separated by semicolons ';' and arranged from left to right. A horizontal line segment is described by a pair of values separated by a comma ','. The first value describes the height of this line segment (how high it is from the ground), and the second value describes the length of this line segments (on the horizontal dimension). The heights of two consecutive line segments should be different. Heights of line segments should be formatted into 3 decimal places after decimal point, and the lengths of line segments should be integers. There is a space after a semicolon or a comma, but there is no space at the end of each output line.

Sample Output

```
29.500, 120; 60.000, 20; 52.800, 100
18.667, 120; 60.000, 20; 41.800, 100
43.500, 120; 60.000, 120
71.750, 240
```

The illustration of skyline with rainfall 20 are shown below. The water level to the left of the tallest building is at 29.5 meters, and the length on the horizontal dimension is 120 meters. The water level to the right of the tallest building is 52.8 meters, and the length on the horizontal dimension is 100 meters. Therefore, the skyline is described by 3 line segments: line segment (29.500, 120), line segment (60.000, 20) and line segment (52.800, 100)



Hint: You need to calculate the minimum rainfall that changes the skyline. For example, when there is a rainfall of 0.8, the building of height 28 will be flooded to the height of 30, so the two building of height 30 and height 28 will have the same water level from then on. After the rainfall of 0.8, if there is another rainfall of 0.2 (accumulated rainfall of 1), the empty ground will be leveled up with the building beside of height 5. You will need to write a loop over this change of skyline, and the flooded skyline will be set after the accumulated rainfall reaches the given amount.

[Question 1.34] Lambda expressions for two sorting requirements (5 marks)

Write a lambda expression to sort words by their length, and then by the alphabetical order. You should assume there is no special characters in the words, and all words are in lower cases.

Input & Output

Words to be sorted may span across multiple lines in the input. You should read all lines from the input file and split each line into words. The total number of words does not exceed 1000. Words are all in lower cases, and there is no duplicate for the same word. You should sort all the words by the two criteria specified above and print one word per line in the sorted sequence.

Sample Input

```
conjugate continuous
convergence
convex conjunction
congruent concrete
conjecture concave
consecutive constant concentric
```

Sample Output

```
convex
concave
concrete
constant
congruent
conjugate
concentric
conjecture
continuous
conjunction
consecutive
convergence
```

Grading Criteria:

1. This is an individual assignment. You are not allowed to discuss among yourselves on the implementation details, you should confine your discussion within high-level hints and directions towards potential solutions. Once submissions are identified as plagiaristic copies, all submissions will be awarded with zero marks as the penalty. We urge students not to disclose your solution to anyone else.
2. For each question, 10% of the marks will be awarded if the expected sample output is produced, even if the sample output is hard-coded.
3. There should be 5 sets of test cases for each question, which are similar to the test case and follow the input specifications described in each question.
4. A program fails to pass a set of test cases if either it does not halt in 3 seconds, or it does not produce the expected output.
5. If the submitted program passes all 5 test cases, 100% of the marks will be awarded.
6. If the submitted program passed 4 test cases, 80% of the marks will be awarded.
7. If the submitted program passed 2 or 3 test cases, 20% to 50% of the marks should be awarded depending on the thinking, the effort and the interpretability of the assignment judged by the grading instructors.
8. The total mark of an assignment is rounded down to the nearest integer.