Q1. Ohm's Law: Voltage, Current, and Resistance (30 marks)

Ohm's Law is a fundamental principle in electrical circuits that relates the voltage (V), current (I), and resistance (R) in a conductor. Mathematically, it can be expressed as:

$$V = I \cdot R$$

where:
V represents the voltage across the conductor (in volts).
I is the current flowing through the conductor (in amperes).
R is the resistance of the conductor (in ohms).

Write a program to

Input, in sequence:

(1) The voltage across the conductor, V (in volts), where $0 < V \leqslant 150$;
(2) The resistance of the conductor, R (in ohms), where $0.1 \leqslant R \leqslant 100$.

Output:

The current flowing through the conductor, I (in amperes), by Ohm's Law. Round the output value to two decimal places, including any necessary trailing zeros.

试题 1. 欧姆定律：电压、电流和电阻（30 分）

欧姆定律是电路中的一项基本原理，它表明了导体中电压（V）、电流（I）和电阻（R）的关系。数学上，欧姆定律可以表达为：

$$V = I \cdot R$$

其中：
V 代表导体上的电压（以伏特为单位）。
I 是流经导体的电流（以安培为单位）。
R 是导体的电阻（以欧姆为单位）。

试写一程式以

依序输入：

(1) 导体上的电压 V（以伏特为单位），其中 $0 < V \leqslant 150$;
(2) 导体的电阻 R（以欧姆为单位），其中 $0.1 \leqslant R \leqslant 100$。

输出：

根据欧姆定律计算流经导体的电流 I（以安培为单位）。将此输出值近似到两位小数，包括任何必要的尾随零。

Examples (例子)

Input (輸入)	Output (輸出)
120 20	6.00
3.3 0.1	33.00
5 0.24	20.83
1 17	0.06
85 0.7	121.43

Q2. GCD and LCM (30 marks)

The **Greatest Common Divisor (GCD)** of two or more integers is the largest positive integer that divides all of them without leaving a remainder. On the other hand, the **Least Common Multiple (LCM)** of two or more integers is the smallest positive integer that is divisible by all of them. In this question, you are required to find the GCD and LCM of a group of positive integers.

Write a program to

Input, in sequence:

(1) N, the number of positive integers to be considered, where $2 \leq N \leq 5$.
(2) N positive integers, and none of them exceeds 1000.

Output, in sequence:

(1) The GCD of these N integers
(2) The LCM of these N integers

试题 2. 最大公约数和最小公倍数 （30 分）

最大公约数（GCD）是能够整除两个或更多正整数的的最大正整数；而最小公倍数（LCM）则是能够被它们所有整除的最小正整数。在这个问题中，您需要计算一组正整数的最大公约数和最小公倍数。

试写一程式以

依序输入：

(1) N，正整数数量，其中 $2 \leq N \leq 5$；
(2) N 个不超过 1000 的正整数。

依序输出：

(1) 此 N 个数的最大公约数（GCD）
(2) 此 N 个数的最小公倍数（LCM）

Examples (例子)

Input (输入)	Output (输出)
2 100 100	100 100
5 1000 999 997 991 989	1 976181544297000
3 55 455 390	5 30030
4 4 8 2 10	2 40
5 357 969 816 153 561	51 3581424

Q3. Snail and Well (30 marks)

There is a well with a depth of h meters. A snail is crawling upwards from the bottom of the well. During the day, the snail can crawl up m meters, but at night, it slips down n meters. It is known that $0 \leq n < m \leq h \leq 20$. Assume the snail can reach the top of the well on the d-th day. What is the value of d?

Write a program to

Input, in sequence:
Three positive integers, h, m, and n, where their meanings are as defined above. It is also known that $0 \leq n < m \leq h \leq 20$.

Output:
d, indicating that the snail can reach the top of the well on the d-th day, where d is a positive integer.

试题 3. 蜗牛与井 （30 分）

一口井，深 h 米，有一只蜗牛在井底往上爬。白天蜗牛能往上爬 m 米，到了晚上它却滑落 n 米，已知 $0 \leq n < m \leq h \leq 20$。假设蜗牛在第 d 天可以抵达井口，请问 d 的值为何？

试写一程式以

依序输入：
三个正整数，h、m、以及 n，其意如上所定义，并且已知 $0 \leq n < m \leq h \leq 20$。

输出：
d, 意即第 d 天蜗牛可以抵达井口，其中 d 为一个正整数。

Example (例子)

Input (输入)	Output (输出)
20 5 0	4
3 3 2	1
8 7 4	2
6 2 1	5
14 6 3	4

Q4. Anagram Checker (30 marks)

Anagrams are words or phrases formed by rearranging the letters of another word or phrase. For example, the words "**stressed**" and "**desserts**" consist of the same letters but just in different sequences, so they are anagrams. For another example, the phrases "**a gentleman**" and "**elegant man**" are also anagrams. In this question, you are required to check whether two input strings are anagrams, disregarding the space(s), if any, between the words in the strings.

Write a programme to

Input, in sequence:

(1) The first string that contains a word or a phrase.
(2) The second string that contains a word or a phrase.

Output:

"anagram" if the first and second strings are anagrams;
"not anagram" if the first and second strings are not anagrams.

试题 4. 变位词检查器（30 分）

变位词（Anagrams）是由通过重新排列另一个单词或短语的字母而形成的单词或短语。例如，单词"**stressed**"和"**desserts**"由相同的字母组成，只是顺序不同，所以它们是变位词。再举一个例子，短语"**a gentleman**"和"**elegant man**"也是变位词。在这个问题中，您需要检查两个输入字符串是否是变位词，其中字符串中的空格（如有的话）可忽略。

试写一程式以

依序输入：

(1) 第一个包含单词或短语的字符串；
(2) 第二个包含单词或短语的字符串；

输出：

"anagram"，如果第一和第二个字符串是变位词
"not anagram"，如果第一和第二个字符串不是变位词

Examples (例子)

Input (输入)	Output (输出)
listen silent	anagram
eleven plus two twelve plus one	anagram
hello world	not anagram
program algorithm	not anagram
funeral real fun	anagram

Q5. Determining Palindromes (40 marks)

Palindromes are words, numbers, or sentences that read the same backwards and forwards, regardless of spaces, special characters, punctuations, and letter casing. Examples include the word 'madam', the number '404404' and the sentence 'Mr. Owl ate my metal worm'.
Write a programme to determine whether the input string is a palindrome word, a palindrome number, or a palindrome sentence.
Hint: "aba" is considered as a palindrome word and "a b a" is considered as a palindrome sentence.

Write a programme to

Input:
A string.
Please note this string contains at least one letter or one number, but letters and numbers will not appear at the same time.

Output:
(1) Display "1" if the input is a palindrome; display "0" if it is not a palindrome.
(2) If the first output is "1", display "word" if it is a palindrome word, "number" if it is a palindrome number, or "sentence" if it is a palindrome sentence.

试题 5：判断回文（40 分）

回文（Palindromes）是前后读取相同的单词、数字或句子，其中空格、特殊字符、标点符号和字母大小写皆可忽略。回文的例子包括了单词"madam"、数字"404404"和句子"Mr. Owl ate my metal worm"。
试写一程式来判断输入字符串是否是回文单词（palindrome word）、回文数字（palindrome number）还是回文句子（palindrome sentence）。
提示："aba" 是一个回文单词，而 "a b a" 是一个回文句子。

试写一程式以

输入：
一个字符串（string）。
请注意，此字符串至少包含一个字母或一个数字，但字母和数字不会同时出现。

输出：
（1）如果输入是回文，则显示"1"；反之，若不是回文，则显示"0"。
（2）如果以上输出是"1"，若这是回文单词就显示"word"，若这是回文数字就显示"number"，若这是回文句字就显示"sentence"。

Examples (例子)

Input (输入)	Output (输出)
Civic	1 word
Be happy for this moment.	0
A man, a plan, a canal – Panama.	1 sentence
$123,321	1 number
22022022	1 number

Q6. Combination of Multiplication (40 marks)

Given a target number and a list of distinct positive integers, your program is required to find all possible combinations of multiplication using only the provided integers, which result in a product equal to the target number.

For example, given that the target number is **24**, and the **list** of positive integers is,

2 8 12 4 6 3

Then all the possible combinations of multiplication using only the provided integers, which result in a product equal to the targeted number (i.e., 24 in this case) are:

2 2 2 3 → combination 1

2 2 6 → combination 2

2 12 → combination 3

2 4 3 → combination 4

8 3 → combination 5

4 6 → combination 6

Note:

(1) The same integer may be chosen from the list an unlimited number of times.

(2) Two combinations may consist of the same set of integers, but the two combinations are considered as different if the frequency of at least one of the integers is different. For example, 2×2×3×3×6 and 2×3×6×6 are considered two different combinations.

Your program will then be required to compute the sum of all the numbers in each of the combinations. For the above example, the sums computed are as follows:

Combination 1 → The sum is 9

Combination 2 → The sum is 10

Combination 3 → The sum is 14

Combination 4 → The sum is 9

Combination 5 → The sum is 11

Combination 6 → The sum is 10

Finally, your program needs to sort all the sums in ascending order as given below and output the result:

9 9 10 10 11 14

Write a programme to

Input, in sequence:

(1) N, the number of integers in the list, where $2 \leq N \leq 15$.
(2) N distinct integers in the list, where all of them are in the range of [2, 100].
(3) The target number T, where $6 \leq T \leq 300$, and it is greater than all the integers in the list.

Output:

The sums in ascending order, where each of them is the sum of all the integers used in a combination of multiplication(s) that result(s) in a product equal to T.
Your program needs to output "**0**" if there is no combination available.

试题 6. 乘法的组合 (40 分)

给定一个目标数字和一组不同的正整数，您的程式需要使用提供的整数，找到所有可能的乘法组合使得其乘积等于该目标数字。

例如，假设目标数字是 **24**，同时所提供的正整数为，

2 8 12 4 6 3

那么，仅使用所提供的整数并使得乘积等于目标数字（在此例中即 24）的所有可能的乘法组合如下：

2 2 2 3 → 组合 1

2 2 6 → 组合 2

2 12 → 组合 3

2 4 3 → 组合 4

8 3 → 组合 5

4 6 → 组合 6

注意：

(1) 同一个整数可被选取的次数没有限制。

(2) 如果两个组合使用相同的整数，但只要至少一个整数的频率不同，那么这两个组合被视为不同的组合。例如：2×2×3×3×6 和 2×3×6×6 是不同的组合。

然后，您的程式将需要计算每个组合中所有整数的和。对于上面的例子，计算得到的和如下所示：

组合 1 → 其和为 9

组合 2 → 其和为 10

组合 3 → 其和为 14

组合 4 → 其和为 9

组合 5 → 其和为 11

组合 6 → 其和为 10

最后，您的程式需要由小到大的将所有的和排序，并输出结果：

9 9 10 10 11 14

试写一程式以

依序输入：

(1) N, 给定整数组中的整数数量，其中 $2 \leq N \leq 15$。
(2) N 个在此整数组中不同的整数，其取值范围为 $[2, 100]$。
(3) 目标数字 T，其中 $6 \leq T \leq 300$，并且它大于所有整数组中的整数。

输出：

找出所有整数组中可以乘积出目标数字的组合，然后找出每个组合中的整数之和，再按升序（由小到大）排列并输出这些和。
如果整数组中没有相关的乘法组合，您的程式需要输出"0"。

Examples (例子)

Input (输入)	Output (输出)
6 2 8 12 4 6 3 24	9 9 10 10 11 14
5 2 3 4 6 8 24	9 9 10 10 11
4 2 3 4 5 16	8 8 8
3 2 4 5 18	0
3 2 4 5 20	9 9

Q7. Best Delivery Person (40 marks)

A food delivery company assesses the performance of its delivery personnel every week, in order to determine the best delivery person based on the highest single-day earnings achieved by each delivery person within that week.

As per company's policy,
1. Delivery personnel can independently decide their preferred working days and the number of deliveries they will do on each of their working days.

2. The number of deliveries made by a delivery person on a day may vary, ranging from 0 to 10 deliveries.

3. The delivery record of each delivery person in a given week will be stored in the system according to their daily earnings, that is, the sequence of daily earnings will be presented as "Monday/Tuesday/Wednesday/Thursday/Friday/Saturday/Sunday".

4. For example, if the record of a delivery person is,
"**1.45;0.60/10.30;6.35;3.21/1.99;2.88/0/0/0/0**", it shows the following information:
(i) The corresponding delivery person worked continuously for three days from Monday to Wednesday, using a forward slash ('/') to separate the days of deliveries.
(ii) Furthermore, the semicolon (";") is used to separate the delivery fees per each delivery for that particular day. For instance, "1.45;0.60" implies the delivery person completed two deliveries on Monday, and the amounts earned are 1.45 and 0.60, respectively; while "10.30;6.35;3.21" means that the delivery person completed three deliveries on Tuesday, with the amounts earned being 10.30, 6.35, and 3.21, respectively.

5. It is important to note that there may be days where no deliveries were made, and as such, these days are indicated explicitly with a zero. For example, in the given record of "**2.35;1.85/8.55;5.10;3.66/2.90;5.50/0/1.22;3.00/0/0**", it is evident that there were no deliveries made by the delivery person on Thursday, Saturday, and Sunday.

Write a program to

Input, in sequence:
(1) The total number of delivery personnel, N, within the company, where $1 \leq N \leq 10$.
(2) N lines of delivery records of the company's personnel in a given week.

NOTE: Assume that (i) all non-zero single-day earnings are different; (ii) there will not be an all-zero record; and (iii) a maximum of 10 deliveries can be made per person per day.

Output, in sequence:
(1) The delivery record, n, with the highest-single-day earnings among all, given $1 \leq n \leq N$.
(2) The day when the abovementioned highest-single-day earnings occurred. This day must be one of the following: "Monday", "Tuesday", "Wednesday", "Thursday", "Friday", "Saturday", or "Sunday";
(3) The total number of deliveries made by the corresponding delivery person on the day of the highest-single-day earnings;
(4) The total amount of delivery fees earned by the delivery person on the day of the highest-single-day earnings. Display this amount with exactly two decimal points.

试题 7. 最佳送餐员（40 分）

一家送餐公司每周都会根据每位送餐员当周内最高单日收入来评估送餐员的表现，并以此决定最佳送餐员。

以下是公司的政策：
1. 送餐员可以独立决定他们偏好的工作日以及在每个工作日上将会完成的送餐数量。

2. 送餐员每天送餐的数量可能会有所变化，其范围从 0 次到 10 次不等。

3. 在一个给定的星期内，送餐员的送餐记录将根据此星期每一天的格式存储在系统中，即 "Monday/Tuesday/Wednesday/Thursday/Friday/Saturday/Sunday"。

4. 例如，某送餐员的记录为 **"1.45;0.60/10.30;6.35;3.21/1.99;2.88/0/0/0/0"**，则此条记录显示了以下信息：
(i) 送餐员从星期一到星期三连续工作了三天，用正斜杠（"/"）来分隔送餐的日子。
(ii) 此外，分号（";"）用于分隔该特定日子每次的送餐费。例如，"1.45;0.60"表示送餐员在星期一完成了两次送餐，所赚金额分别为 1.45 和 0.60，而"10.30;6.35;3.21"表示送餐员在星期二完成了三次送餐，所赚金额分别为 10.30、6.35 和 3.21。

5. 需要注意的是，在某些日子若该送餐员并没有提供送餐服务，则这些日子会明确地以零表示。例如，若给定的记录为 **"2.35;1.85/8.55;5.10;3.66/2.90;5.50/0/1.22;3.00/0/0"**，则其显示了星期四、星期六和星期天该送餐员没有进行送餐服务。

试写一程式以

依序输入：
（1）公司内的送餐人员总数，N，其中$1 \le N \le 10$。
（2）N行记录，其中每一行对应到公司某一位送餐员在该周内的送餐记录。
注意：我们假设（i）所有非零的单日收入都互不相同；（ii）没有全零的一行记录；（iii）每人每天最多可以进行 10 次送餐服务。

依序输出：
(1) 所有送餐记录中，拥有单日收入最高的送餐记录，假设该记录为 n，则$1 \le n \le N$；
(2) 最高单日收入出现的日子，该日子必须严格为以下格式之一："Monday"，"Tuesday"，"Wednesday"，"Thursday"，"Friday"，"Saturday"，或 "Sunday"；
(3) 在最高单日收入上，该相应送餐员完成的送餐数量；
(4) 该送餐员在最高单日收入的日子上所赚取的送餐费用的总数。此总数必须以两位小数的形式来呈现。

Examples (例子)

Input (输入)	Output (输出)
1 1.45;0.60/10.30;6.35/1.99;2.88/0/0/0/0	1 Tuesday 2 16.65
2 1.45;0.60/5.30;6.35/1.99;2.88/0/0/0/0 1.25;10.60/5.40;1.20/1.80;4.00/0/0/0/0	2 Monday 2 11.85
2 1.45;0.60/10.30;6.35;3.21/1.99;2.88/0/0/0/0 1.25;10.60/5.40;1.20/1.80;7.45/0/0/2.45;1.55/3.50;5.55	1 Tuesday 3 19.86
3 1.45;0.60/10.30;6.35;3.21/1.99;2.88/5.00;3.90/1.00;2.00/3.00;4.00/9.45;8.00 1.25;10.60/5.40;1.20/1.80;7.45/0/0/0/0 2.45;8.60/7.30;4.35;7.21/3.99;5.88/0/1.22;3.00;4.00;5.00;6.00;7.00/0/0	3 Friday 6 26.22
4 1.25;10.60/5.40;1.20/1.80;7.45/0/0/0/0 1.45;0.60/10.30/1.99;2.88/5.00;3.90/2.00/3.00;4.00/9.45;8.00 1.25;10.60/5.40;1.20/1.80;7.45/0/0/2.45;1.55/3.50;5.55 1.45;0.60/10.30;6.35;3.21/0/0/0/0/9.45;8.00;1.00;2.00;3.00;4.00	4 Sunday 6 27.45

Q8. Area of Convex Quadrilateral (40 Marks)

Find the area of the convex quadrilateral bounded by 4 straight lines connecting four given coordinates. For example, given the four coordinates (1,3), (2,2), (3,4), and (-1, -2) as shown in Figure Q8, the area of the convex quadrilateral formed by these coordinates is 5. **However, be caution, you need to arrange the coordinates correctly to form a convex quadrilateral.**

Hint 1: To calculate the distance between two coordinates.
The formula to find the distance between the two points is given by.

$$distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Hint 2: To calculate the area of a triangle using three side lengths.
In geometry, Heron's formula gives the area of a triangle in terms of the three side lengths a, b, and c. Just use this two-step process:

Step 1: Calculate "s" (half of the triangles perimeter):

$$s = \frac{a + b + c}{2}$$

Step 2: Then calculate the Area:

$$Area = \sqrt{s(s - a)(s - b)(s - c)}$$

Finally, a convex quadrilateral consists of 2 triangles.

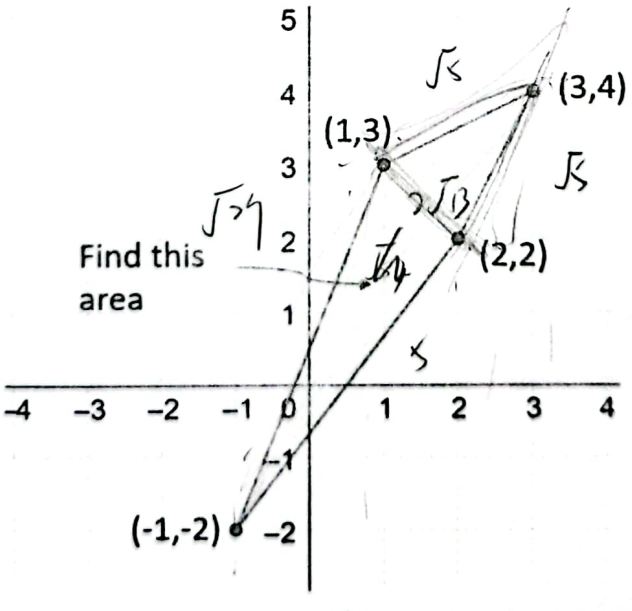


Figure Q8. The area of quadrilateral formed by four coordinates.

Write a program to

Input, in sequence:
Four coordinates in the form of

x_1 y_1

x_2 y_2

x_3 y_3

x_4 y_4

Note: All the values of the above coordinates are integers in the interval [-100, 100].

Output:
The area of convex quadrilateral formed by the above coordinates.

试题 8. 凸四边形的面积（40 分）

给定四个坐标，假设它们的连线可以界定一个凸四边形（convex quadrilateral），求出这个四边形的面积。例如，给定坐标(1, 3)、(2, 2)、(3,4) 和 (-1, -2)，如图 Q8 所示，由这些坐标形成的凸四边形的面积为 5。**然而，需要注意的是，您需要正确排列这些坐标以形成一个凸四边形。**

提示 1：计算两个坐标之间的距离。

计算两点之间距离的公式如下：

$$\text{距离} = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

提示 2：使用三边长度计算三角形的面积。

在几何中，海伦公式（Heron's formula）能按照以下两步计算，利用三条边长 a、b、和 c 为参数，找出三角形的面积。：

步骤 1：计算"s"（三角形半周长）：

$$s = \frac{a + b + c}{2}$$

步骤 2：然后计算面积：

$$\text{面积} = \sqrt{s(s-a)(s-b)(s-c)}$$

最后，凸四边形由 2 个三角形组成。

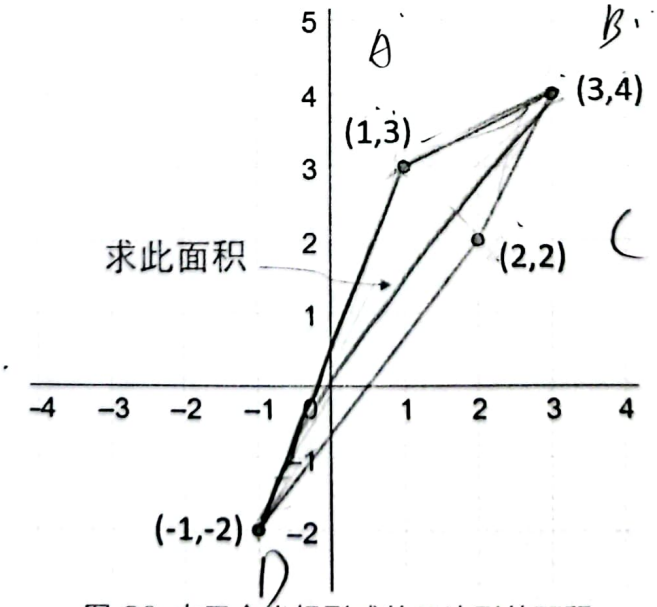


图 Q8. 由四个坐标形成的四边形的面积

试写一程式以

依序输入:
以下形式的四点坐标

x_1 y_1

x_2 y_2

x_3 y_3

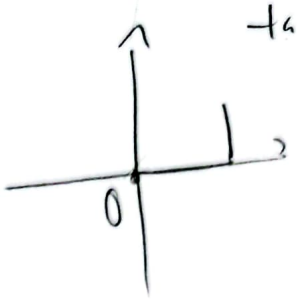
x_4 y_4

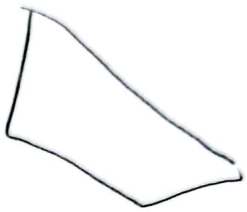
注意：以上坐标的数值皆为整数，且都落在区间 [-100, 100] 里。

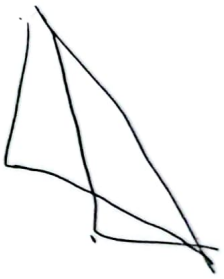
输出:
由这四个坐标形成的凸四边形的面积。

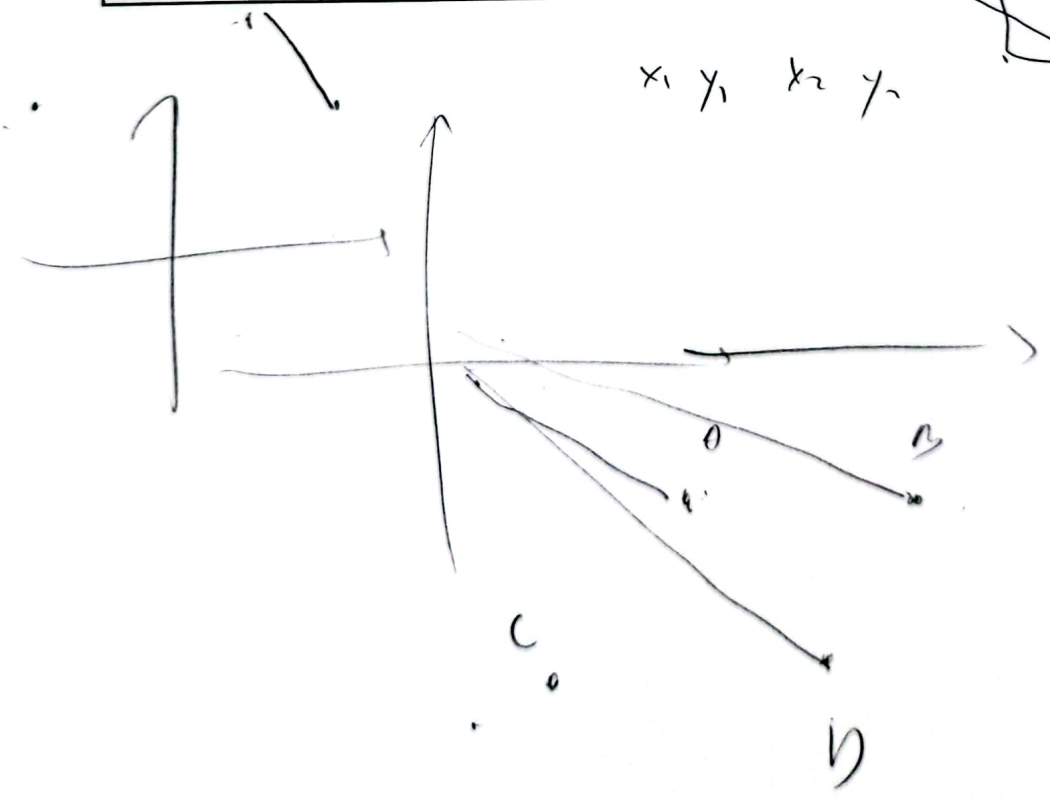
Examples (例子)

Input (輸入)	Output (輸出)
2 2 3 4 1 3 -1 -2	5
22 40 47 40 9 23 30 23	391
10 7 -7 8 5 17 7 -8	211.5
18 10 14 46 7 15 51 29	820
21 -4 50 -4 12 -10 37 -10	162









Q9. Edit Distance (50 marks)

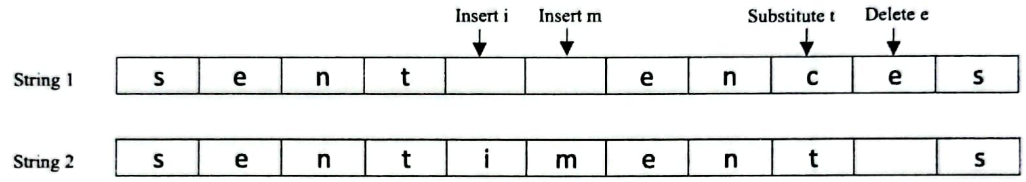
Consider two English words. The **edit distance** between the two words is defined as the minimum number of changes to be made in order to transform one word to the other word. The changes may include substitution(s), deletion(s), and/or insertion(s).

For example, if the two words are "sentences" and "sentiments". With reference to Figure Q9, in order to transform the word "sentences" to the word "sentiments", one will need to

(i) substitute the "c" at the third last position with a "t" (i.e., one substitution);
(ii) delete the "e" at the second last position (i.e., one deletion), and
(iii) insert an "i" and an "m", respectively (i.e., two insertions), as shown in the Figure.

Thus, the edit distance between "sentences" and "sentiments" is 1+1+2 = 4.

Please note that the cases of the letters can be ignored.



Number of Substitution(s): 1
Number of Deletion(s) 1
Number of Insertion(s) 2
Edit Distance 1+1+2 = 4

Figure Q9. The edit distance between "sentences" and "sentiments"

Write a programme to

Input:
Two words.

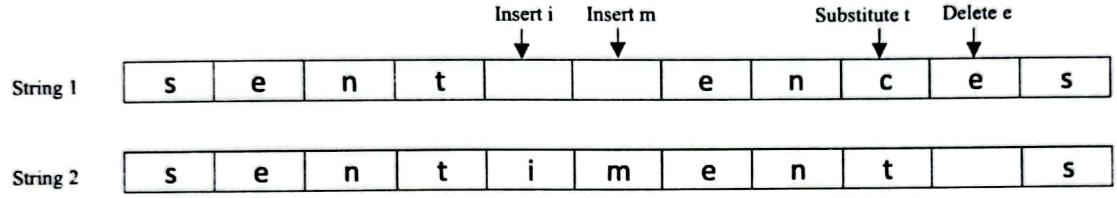
Output:
The edit distance, *d* between the two words.

Q9. Edit Distance (50 marks)

Consider two English words. The **edit distance** between the two words is defined as the minimum number of changes to be made in order to transform one word to the other word. The changes may include substitution(s), deletion(s), and/or insertion(s).
For example, if the two words are "sentences" and "sentiments". With reference to Figure Q9, in order to transform the word "sentences" to the word "sentiments", one will need to

(i) substitute the "c" at the third last position with a "t" (i.e., one substitution);
(ii) delete the "e" at the second last position (i.e., one deletion), and
(iii) insert an "i" and an "m", respectively (i.e., two insertions), as shown in the Figure.

Thus, the edit distance between "sentences" and "sentiments" is 1+1+2 = 4.
Please note that the cases of the letters can be ignored.



Number of Substitution(s): 1
Number of Deletion(s) 1
Number of Insertion(s) 2
Edit Distance 1+1+2 = 4

Figure Q9. The edit distance between "sentences" and "sentiments"

Write a programme to

Input:
Two words.

Output:
The edit distance, *d* between the two words.

试题 9. 编辑距离（50 分）

考虑两个英文单词。这两个单词之间的**编辑距离**（edit distance）可定义为将其中一个单词转换为另一个单词所需进行的最小更改次数。更改可能包括替换、删除和/或插入。例如，如果两个单词是"sentences"和"sentiments"。参考图 Q9，为了将单词 "sentences" 转换为单词 "sentiments"，您需要：

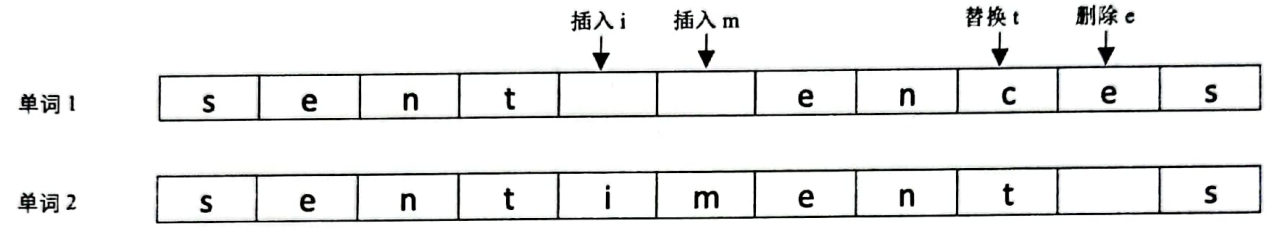
(i) 将倒数第三个位置的 "c" 替换为 "t"（即 1 个替换）；

(ii) 删除倒数第二个位置的 "e"（即 1 个删除）；

(iii) 分别插入 "i" 和 "m"（即 2 个插入），如图中所示。

因此，"sentences"和"sentiments"之间的编辑距离是 1+1+2 = 4。

请注意，字母的大小写可以忽略。



替换次数: 1
删除次数: 1
插入次数: 2
编辑距离: 1+1+2=4

图 Q9. "sentences" 与 "sentiments" 之间的编辑距离

试写一程式以

输入：

两个单词。

输出：

两个词之间的编辑距离，d。

Examples (例子)

Input (输入)	Output (输出)
kitten kitchen	2
park phase	3
cat dog	3
computer laptop	6
grill GriLL	0
length height	4
File Fail	2

Q10. Dish Cooking Time (50 marks)

In a restaurant kitchen, there are various dishes that need to be prepared by a team of chefs. Each dish has a specific cooking time, and the kitchen can accommodate the cooking of up to K dishes concurrently.

However, when multiple dishes are being cooked at the same time, all the ongoing cooking processes must be finished before new dishes can be started. For example, if Dish A takes 10 minutes to cook and Dish B takes 15 minutes, the concurrent cooking of dishes A and B would require 15 minutes for the chefs to complete, and only then can they proceed to cook the next dish(es).

Given a list of dishes with their respective cooking times (in minutes) and the parameter K, representing the maximum number of dishes that can be cooked concurrently, your objective is to calculate the minimum time (in minutes) required for all the dishes to be fully prepared in the kitchen.

Note: The team of chefs can choose any sequence to cook the dishes.

Write a program to

Input, in sequence:

(1) a positive integer N, which indicates the number of dishes to be cooked, where $2 \leq N \leq 20$;
(2) N positive integers, each not exceeding 20, representing the cooking times (in minutes) of the N dishes, respectively;
(3) a positive integer K, which indicates the maximum number of dishes that can be cooked concurrently, where $1 \leq K \leq 5$.

Output:

The minimum time (in minutes) required for the kitchen to complete cooking all the dishes.

试题 10. 菜品烹饪时间（50 分）

在一家餐厅的厨房里，有各种不同的菜品需要由一组厨师准备。每道菜品都有特定的烹饪时间，而厨房可以同时烹饪最多 K 道的菜品。

然而，当多个菜品同时被烹饪时，所有正在进行的烹饪过程必须完成，然后厨师们才能开始烹饪下一道（或下几道）菜品。例如，如果菜品 A 需要烹饪 10 分钟，而菜品 B 需要烹饪 15 分钟，在同时开始烹饪菜品 A 和 B 的情况下，厨师们需要 15 分钟来完成这两道菜品的烹饪，然后他们才能开始下一道（或下几道）菜品的烹饪。

给定一份菜品清单，其中包括每道菜品的烹饪时间（以分钟为单位）以及参数 K，表示厨师们可以同时烹饪最多菜品的数量，您的目标是计算厨房完成所有菜品所需的最短时间（以分钟为单位）。

注意：厨师团队可以选择任何顺序来烹饪这些菜品。

试写一程式以

依序输入:

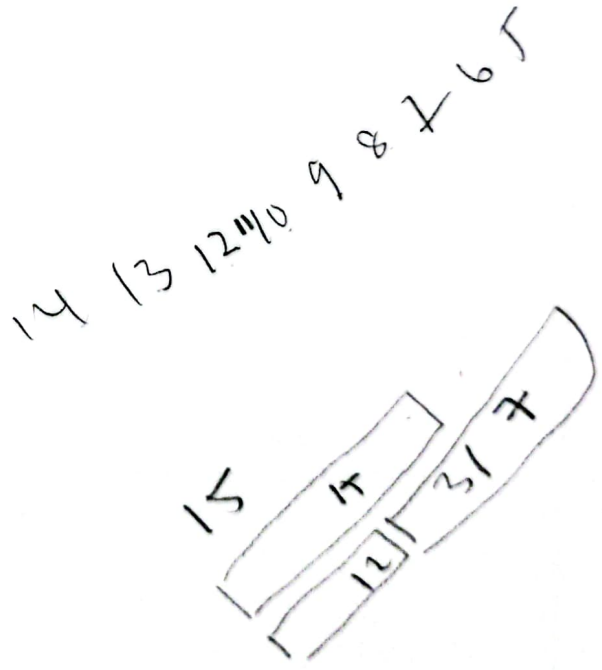
(1) 一个正整数 N，表示要烹饪的菜品的数量，其中$2 \leq N \leq 20$;
(2) N个不超过 20 的正整数，分别表示 N 个菜品的烹饪时间（以分钟为单位）；
(3) 一个正整数 K，表示厨房可以同时烹饪菜品最多的数量，其中 $1 \leq K \leq 5$。

输出:

厨房完成烹饪所有菜品所需的最短时间（以分钟为单位）。

Examples (例子)

Input (输入)	Output (输出)
10 9 11 14 8 13 7 6 12 10 5 3	38
3 10 15 12 2	25
12 15 19 20 9 10 13 14 19 10 9 10 10 3	55
15 11 9 6 13 8 14 5 12 7 10 15 5 13 6 10 4	42
19 1 10 18 7 4 13 6 19 2 16 8 11 3 17 14 9 5 12 20 5	47



Q11. Path Finder (60 marks)

Consider that a robot operates within a 2-dimensional 200 × 200 grid. Each grid cell, say grid cell i, is labelled with a coordinate (x_i, y_i), where $1 \leq x_i, y_i \leq 200$.

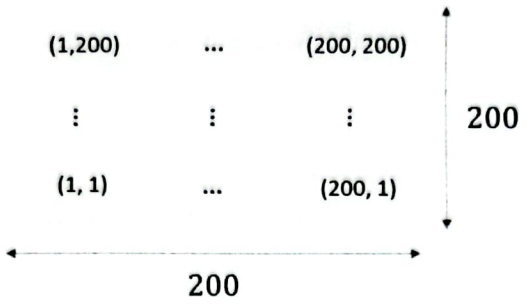


Figure Q11: Grid Environment

The robot will commence its journey at a start point, denoted as (x_0, y_0), and it is aimed to travel to reach a specified target point, (x_t, y_t). Note that both the start point and target point are guaranteed to be within the grid.

However, the robot cannot move arbitrarily, but only execute a series of movements encoded in a set, $\boldsymbol{A}$. That is, each element of $\boldsymbol{A}$ is a pair $(\Delta x, \Delta y)$ that indicates a valid movement of Δx steps in the x-direction and Δy steps in the y-direction, where $-200 \leq \Delta x, \Delta y \leq 200$.

Suppose that **a valid move** is defined as an execution of one of the movements given in $\boldsymbol{A}$, and the robot cannot move outside the grid boundaries. The task is to determine the minimum number of moves the robot is required to make in order to arrive at (x_t, y_t) from (x_0, y_0). Note also each movement in set $\boldsymbol{A}$ can be repeated multiple times if necessary.

If it is impossible for the robot to reach the target point using the movements in set $\boldsymbol{A}$, your program should return -1.

For example:

- **Given that**
  - Initial position: $(x_0, y_0) = (3,3)$
  - Target position: $(x_t, y_t) = (5,5)$
  - Number of distinct movements: $n = 2$
  - Movements in set $\boldsymbol{A}$:
    - Move right by one step: $(\Delta x, \Delta y) = (1,0)$
    - Move up by one step: $(\Delta x, \Delta y) = (0,1)$
- **Solution:**
  - The minimum number of moves the robot needs to take to reach the target positions is 4. That is, move two steps with (0, 1) and two steps with (1, 0). The order of the moves does not matter in this case.

Write a programme to

Input, in sequence:

(1) Initial position x_0 and y_0, where $1 \leq x_0, y_0 \leq 200$, and both are integers.
(2) Target position x_t and y_t, where $1 \leq x_t, y_t \leq 200$, and both are integers.
(3) An integer n, denoting the number of distinct movements the robot can perform ($1 \leq n \leq 10$).
(4) n pairs of integers, each pair denoting a possible movement $(\Delta x, \Delta y)$ in set $\boldsymbol{A}$.

Output:

The minimum number of moves the robot needs to take in order to reach the target position by executing any sequence of the movements in set $\boldsymbol{A}$ and repeating any movement if necessary. If it's impossible for the robot to reach the target position using the movements in set $\boldsymbol{A}$, output -1.

试题 11. 路径搜寻器（60 分）

考虑一个机器人在一个二维的 200 × 200 网格中操作。每个网格单元，例如网格单元 i，都带有一个坐标 (x_i, y_i)，其中$1 \le x_i, y_i \le 200$.

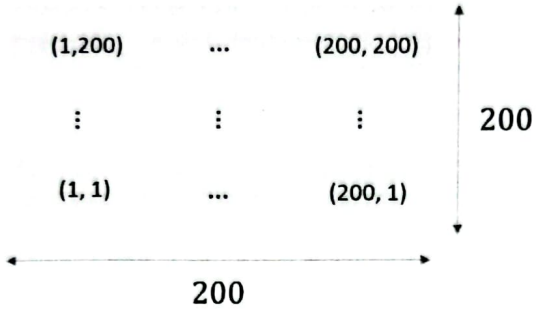


图 Q11: 网格环境

机器人将从一个起点开始它的旅程，且此起点表示为 (x_0, y_0)，而它的目标是前往一个指定的目标点，(x_t, y_t)。注意，起始点和目标点都保证在网格内。

然而，机器人不能任意移动，而只能执行一系列在集合 $\boldsymbol{A}$ 中编好的动作。也就是说，$\boldsymbol{A}$ 的每个元素都是一个移动指令 $(\Delta x, \Delta y)$，表示了当执行此移动指令时，机器人会在x方向上移动 Δx 步，以及在y方向上移动 Δy 步，其中$-200 \le \Delta x, \Delta y \le 200$。

假设**一个有效的移动动作**被定义为执行集合 $\boldsymbol{A}$ 中其中一个移动指令，并且机器人不能移动到网格边界之外。您的任务是，通过使用集合$\boldsymbol{A}$ 中的移动指令，寻找机器人从其初始位置到达指定的目标点的最少可能的动作数目。值得注意的是，如有需要，任何一个移动指令可以被重复多次使用。

如果机器人无法使用集合 $\boldsymbol{A}$ 中的移动指令到达目标点，则您的程序应返回 -1。

例子:

- **倘若:**
  - 初始位置: $(x_0, y_0) = (3,3)$
  - 目标位置: $(x_t, y_t) = (5,5)$
  - 移动指令的数量: $n = 2$
  - 集合 $\boldsymbol{A}$ 中的移动指令:
    - 向右移动一步，即 $(\Delta x, \Delta y) = (1,0)$
    - 向上移动一步，即 $(\Delta x, \Delta y) = (0,1)$
- **解答:**
  - 机器人到达目标位置所需的最少动作次数是 4 次。也就是说，用 $(0, 1)$ 指令移动两次，再用 $(1, 0)$指令移动两次。动作的顺序在此例中并不重要。

试写一程式以

依序输入：

（1）初始位置(x_0, y_0)，其中 $1 \le x_0, y_0 \le 200$，并且它们都是整数。

（2）目标位置(x_t, y_t)，其中 $1 \le x_t, y_t \le 200$，并且它们都是整数。

（3）一个整数 n，即机器人可以执行的不同移动指令的数量 $(1 \le n \le 10)$。

（4）n 对整数，每对表示集合 A 中可能的一个移动指令 $(\Delta x, \Delta y)$。

输出：

机器人抵达目的所需的最少动作次数。机器人必须利用集合 A 中的移动指令来做移动的动作，且若有需要，某个移动指令可重复多次使用。

如果机器人无法使用集合 A 中的移动指令以到达目标位置，则输出 -1。

Examples (例子)

Input (输入)	Output (输出)
1 1 6 6 3 1 1 2 1 2 2	3
10 10 1 1 2 1 0 0 1	-1
1 1 1 1 1 0 0	0
3 3 5 5 2 1 0 0 1	4
5 5 10 10 1 -1 -1	-1

Q12. Code Breaking Challenge (60 marks)

Dena is known for her expertise in deciphering complex codes. She has been assigned a top-secret project that presents her with a challenge — breaking a cryptic code composed of a number string.

The cryptic code, denoted as S, consists of N decimal digits. Dena discovers crucial rules to decipher the code during her investigation: (1) the deciphered code consists of the similar digits as that of the cryptic code, but the digits may be in different sequence; (2) the deciphered code must not contain the "01" substring to unravel the hidden message within. Her primary objective is to break the cryptic code by obtaining a modified string that eliminates all occurrences of "01" substring.

Leveraging her knowledge of code-breaking techniques, Dena further realizes that strategic swaps of adjacent digits hold the key to remove the occurrences of substring "01" from the cryptic number string. For example, a cryptic number string "301" can be transformed with a swap of the second and third digits to make it into "310" to eliminate the occurrence of "01". Minimizing the number of swaps required to eliminate the "01" substrings becomes imperative, as it allows Dena to break the code quickly and unravel the hidden message within.

For example, consider a scenario where $N = 5$ and the initial code string S is "70010".
By swapping the third and fourth digits, the code string transforms into "70100".
Subsequently, by swapping the second and third digits, the code string transforms into "71000".

With the above processes, remarkably, the modified code string no longer contains the "01" substring, aligning with Dena's decoding objective.
In this case, Dena successfully breaks the code by performing a minimum of 2 swaps.

Write a programme to

Input, in sequence:
An integer N denoting the length of the cryptic code (number string), where $1 \leq N \leq 50$;
A cryptic code (number string), S, which consists of N digits.

Output:
The minimum number of swaps required to obtain a modified code string without the "01" substring.

试题 12. 密码破解挑战 (60 分)

迪娜以破解复杂密码而闻名，她被分配到了一个绝密的任务 —— 破解由数字串组成的神秘密码。

这个神秘密码可用 S 表示，由 N 个十进制数字组成。在调查过程中，迪娜发现了破解密码的关键规则: (1) 破解后的密码与神秘密码由相同的数字所组成，但其顺序可能不同; (2) 破解后的密码不能包含子数字串"01"，以解开隐藏在其中的信息。迪娜的主要目标是通过消除神秘密码里所有子数字串"01"，从而获得一个修改后的数字串来破解神秘密码。

凭借她对密码破解技术的了解，迪娜进一步意识到了策略性的交换相邻数字是消除神秘密码中子数字串"01"的关键。例如，神秘数字串"301"可以通过交换第二位和第三位数字变成"310"，从而消除"01"的出现。如此一来，尽量减少所需的交换次数以消除所有子数字串"01"变得非常重要，因为这样可以让迪娜快速破解密码并解开其中隐藏的信息。

例如，考虑 $N = 5$，同时初始密码数字串 S 为"70010"。

通过交换第三位和第四位数字，密码数字串可变为"70100"。

接着，通过交换第二位和第三位数字，密码数字串可变为"71000"。

值得注意的是，经过过程后，修改过的数字串不再包含子数字串"01"，这也达到了迪娜解码的目标。

在上述情况下，迪娜成功地通过了最少的 2 次交换来破解了密码。

试写一程式以

依序输入:

一个正整数 N，以表示神秘密码（数字串）的长度，其中 $1 \le N \le 50$;

一个由 N 个数字所组成的神秘密码（数字串）S。

输出:

消除子数字串"01"所需的最小交换次数，以获得修改后的密码数字串。

Examples (例子)

Input (输入)	Output (输出)
5 70010	2
8 00134001	3
20 01010001020103111001	8
40 2101314151617181912131415161711181912131	1
22 0110101101019110102111	14