A.

Problem : Continuous Niven Numbers

In recreational mathematics, a Niven number in a given number base, is an integer that is divisible by the sum of its digits when written in that base. For example, in base 10, 18 is a Niven number since 18 is divisible by 1+8 = 9. Also, 12001 in base 3 is also a Niven number since the sum of the digits is 4 (which is 11 in base 3) divides 12001 (12001 = 1021 x 11).

Given a base b, any number n < b is trivially a Niven number. We will ignore this case.

Given a base b, and a positive integer T, find the lowest number L such that L, L+1, ..., L+T-1 are all Niven numbers but neither L-1 nor L+T are Niven numbers.

Input Format:

First line contains two integers, b and T

Output Format:

A single integer L such that L, L+1, ..., L+T-1 are all Niven numbers but neither L-1 nor L+T are Niven numbers.

Constraints:

$2 \leq b \leq 10$
$1 < T < 7$

Example 1

Input
10 4

Output
510

Explanation
510, 511, 512 and 513 are Niven numbers and 514 is not a Niven number. Also 509 is not a Niven number. It can be seen that for N < 510, no four consecutive numbers are Niven numbers.

Example 2

Input
5 5

Output
44

Explanation
44 in base 5 is equivalent to 24 in base 10. Clearly, sum of the digits is 8 = 13 in base 5 and 13 x 3 = 44 in base 5 and hence 44 is a Niven number. Similarly we can see 44+1 = 100, 101, 102 and 103 in base 5 are also Niven numbers. 104 is not a Niven number.

B.

Problem : Palindrome sorting

A palindrome is a word, phrase, or sequence that reads the same backwards as forwards. Given a palindrome write a program to print the sorted list of all palindromes that can be constructed from the alphabets of the given palindrome. All palindromes should start in a newline.

Input Format:

The first line, an integer T, indicating the number of test cases

T lines each containing one string (palindrome)

Output Format:

For each test case, one line indicating the Test case number

The next lines containing a sorted list of all palindromes constructed from the given palindrome of the ith test case

Constraints:

$T \leq 10$

Example 1

Input
1
NITIN

Output
INTNI
NITIN

Explanation
There are only two palindromes that can be constructed from NITIN

Example 2

Input
1
ABCDCBA

Output
ABCDCBA
ACBDBCA
BACDCAB
BCADACB
CABDBAC
CBADABC

Explanation
Since there is only one D, it can only be the middle letter in the palindrome. The remaining lettere A, B, C can be arranged in 6 possible ways and each way gives rise to a palindrome.

C.

Problem : Round Vaults in Bank

In the strong room of ABC bank there are N vaults arranged in a circle. The amount of money inside each vault displayed on the door. You can empty any number of vaults as long as you do not empty more than 2 out of any 5 adjacent vaults. If you attempt to break more than 2 of any 5 adjacent vaults, an alarm sounds and the sentry a sharp shooter will kill you instantly with his laser gun! Note that as the vaults are arranged in a circle, the last vault is adjacent to the first one.

The output is the maximum amount of money that can be emptied without sounding the alarm

Input Format:

The first line contains an integer N which is the number of vaults. The next line has a sequence of positive integers of length N, giving the amount of cash in its vaults in order

Output Format:

The maximum amount of money that can be looted without sounding the alarm.

Constraints:

N<=50, Amount in each vault <=50000

Example 1

Input
9
1000, 2000, 1000, 5000, 9000, 5000, 3000, 4000, 1000

Output
15000

Explanation
The vaults 1, 5, 6 are looted, giving a total loot of (1000+5000+9000)=15000

Example 2

Input
10
1000,2000,3000,5000,9000,7000,6000,4000,7000,5000

Output
26000

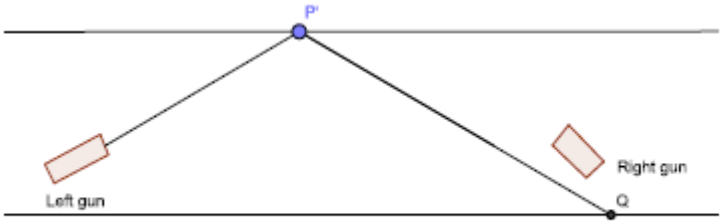
Explanation
There are 10 vaults arranged in a circle. The amounts in the vaults are 1000, 2000, ... 5000.

One way of getting the maximum is to loot vaults 4, 5, 9 and 10 giving a total of 26000. Hence the output is 26000. Note that no 5 adjacent vaults have more than 2 looted.

D.

Problem : Colliding Cannons

We have seen in many mythological movies, the arrows shot, by the opponents collide midair and one devour the other.



You wanted to simulate a similar situation for the video game you are designing. In the game, the opponents are in a tunnel and have a gun each. They can shoot bullets in any direction (within limits). The roof and floor of the tunnel are perfect surfaces and any object hitting them are bounced off according to the law of reflection (angle of incidence equals the angle of reflection), with unchanged speed. For simplicity, we can assume that the tunnel is a two dimensional horizontal strip. Of course, this being the mythological world, gravity does not exist, and the bullets travel in straight lines at constant spped until being reflected (or they collide).

The two guns are positioned at half the height (h) of the tunnel, at a distance D apart.

The two guns fire simultaneously. The trajectories of the bullets (if extended) will meet at a maximum of one point. They are said to collide if their trajectories meet, and the two bullets arrive at that point within 0.5 seconds of each other.

The shooting angle varies from -85 degrees to 85 degrees from the horizontal. For the left gun, the angles are measured anti-clockwise, and for the right gun they are measured clockwise. Hence, with a positive angle for both, the left gun shoots up and to the right, and the right gun shoots up and to the left.

Write a program to decide whether the bullets shot will collide or not. If they do, determine where will they collide. The coordinate axes for reference have origin at the midpoint of the line joining the guns, X axis along the line joining the guns. Hence, the left gun's coordinates are (-D/2,0) and the right gun's coordinates are (D/2,0). The equations for the top of the tunnel is y=h/2, and of the bottom of the tunnel is y=-h/2.

Input Format:

The input has two lines.

The first line has two comma separated positive integers, h and D (the height of the tunnel and the distance between them). The unit of distance measurement is myth units.

The second line has four comma separated numbers (with up to two decimals each) giving LA, LS, RA and RS respectively, where LA, LS denote the angle (in degrees) and speed of firing (in myth units per second) of the left gun and RA, RS denote the angle (in degrees) and speed of firing (in myth units per second) of the right gun.

Output Format:

If the two collide (the trajectories meet, and they arrive at the meeting point within 0.5 seconds of each other), the output is a comma separated string of the word Yes and the coordinates of the colliding point.

Yes, x, y

Here, x and y are the coordinates of the collision point, and each must be rounded to two decimal places If they do not collide, the output is the word No.

No

Constraints:

-85 ≤ LA, RA ≤85
0 < LS, RS <1000
0 < h, D < 10000

Example 1

Input
500,2000
30,90,40,70

Output
Yes,46.20,171.01

Explanation
The distance between the guns is 500 myth units, and the height of the tunnel is 2000 myth units. The angle of firing of the left gun is 30 degrees and the speed of the bullet from the let gun is 90 myth units per second. The angle of the right gun is 40 degrees, and the speed of the bullet from the right gun is 70 myth units per second.

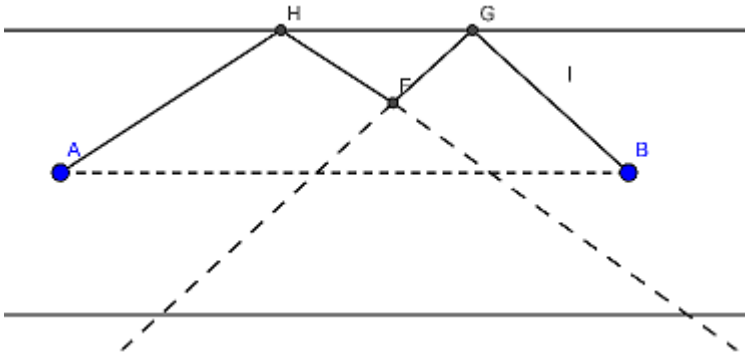
The coordinates of the intersection point of the trajectories is (to two decimal places) (46.20,171.01). The time for the left bullet to reach this point is (to two places) 3.80 seconds, and for the right bullet (to two seconds) is 3.80 seconds. As this is within 0.5 seconds of each other, the collision is assumed to have taken place. Hence the output is Yes,46.20,171.01

Example 2

Input
500,2000
80,70,85,70

Output
Yes,84.19,104.74

Explanation
The distance between the guns is 500 myth units and the height of the tunnel is 2000 myth units. The left gun angle is 80 degrees, and the left gun speed is 70 myth units per second. The right gun angle is 85 degrees and the speed is 70 myth units per second also.



The guns shoot, and reflect off the ceiling (at H and G respectively), and the trajectories meet at F. The coordinates of F are (84.19, 104.74). The time taken for the left bullet is 27.49 seconds, and the time for the second bullet is 27.17 seconds. As they arrive within 0.5 seconds of each other, this is considered a collision. Hence the output is Yes,84.19,104.74

Example 3

Input
500,2000
30,170,50,160

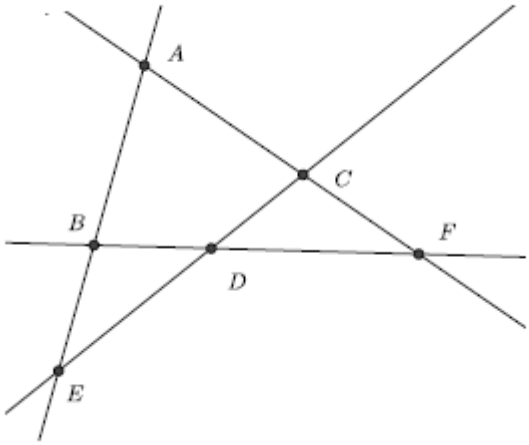
Output
No

Explanation
The two trajectories meet at (86.82,194.47), but the left bullet takes 2.29 seconds, and the right bullet takes 1.59 seconds. Hence, they do not pass within 0.5 seconds of each other, and there is no collision.

E.

Problem : Special Triangles

When we have 4 lines in the plane, the maximum number of triangles formed by them is 4 as shown below: Of the four triangles ABF, CDF, BDE, ACE, triangles BDE and CDF are somewhat "special". They do not have another of the given lines go through their interiors.



Given n lines in the plane, the objective is to count the number of special triangles formed by them. You may assume that no three lines will be concurrent (go through the same point). Each line will be specified by two points (x and y coordinates of two points on the line).

Note that three lines may not form a triangle, let alone a special triangle, if two of them are parallel lines.
Input Format:

The first line will have a single integer, N, representing the number of lines in the input.

The next N lines will have four comma separated numbers representing x1, y1, x2, y2, (the x and y coordinates of two points on the line.)

Output Format:

The output is a single line giving the number of special triangles formed.

Constraints:

N≤20

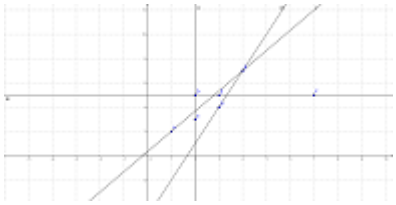
Each of the x and y coordinates of the points defining any line lies between -10 and 10.

Example 1

Input
4
1,2,4,7
2,3,2,5
3,5,7,5
4,7,3,4

Output
2

Explanation



Line I is defined by the points (1,2) (A) and (4,7) (B). Similarly, line II is defined by C (2,3) and D (2,5), line III by E (3,5) and F(7,5) and line IV by B (4,7) and G (3,4).

Of the four triangles formed by the four lines, the ones by (I,II,III) and (I,III,IV) are special.

(I,II,IV) has III going through the interior, while (II,III,IV) has I going through the interior.

As only two triangles are special, the output is 2.

Example 2

Input
4
2,2,3,2
2,3,3,3
2,4,3,4
2,5,3,5

Output
0

Explanation
As all lines are parallel to the x axis, there are no triangles, and hence no special triangles. The output is 0.

F.

Problem : Race in Space

This is the year 3017. Humans have conquered the galaxies and invented new forms of travel. The concept of time has taken new meaning, and spaceships can travel between stars very quickly.

A new discovery is that of wormholes, which exist between some stars, which can take you back in time. If you enter a wormhole at time T, you exit it at the other point at time FLOOR(T/2). Here the FLOOR(x) gets the largest integer which is not greater than x.

Stars are numbered 1 to N, and star catalogues are available online. This contains time taken for travel from one star to another. Note that due to the position of black holes, the time given for travel from Star I to Star j is for that direction only, and travel in the reverse direction may take a different length of time, or not possible at all. If a path from Star I to Star j is not listed, it is not possible to travel between these two directly.

Worm holes may also be travelled in one direction only, and are indicated with a travel time of -1

The objective is to determine the earliest time to arrive at a given destination from a given start point, assuming you are starting at time T=0.

Input Format:

The first line is a pair of space delimited positive numbers giving the starting star number and the destination star number

The next line is a pair of positive integers giving the total number of stars (N) and the number of star paths and worm holes (k) The next k lines specify the star paths and worm holes, and have three space separated numbers giving the origin and the destination star numbers, and the time taken for the travel respectively. If it is a wormhole, the travel time is indicated as -1

Output Format:

The output is a single integer on a line indicating the earliest time by which the destination can be reached.

Constraints:
br>N≤20
K≤50
Time for each path ≤1000

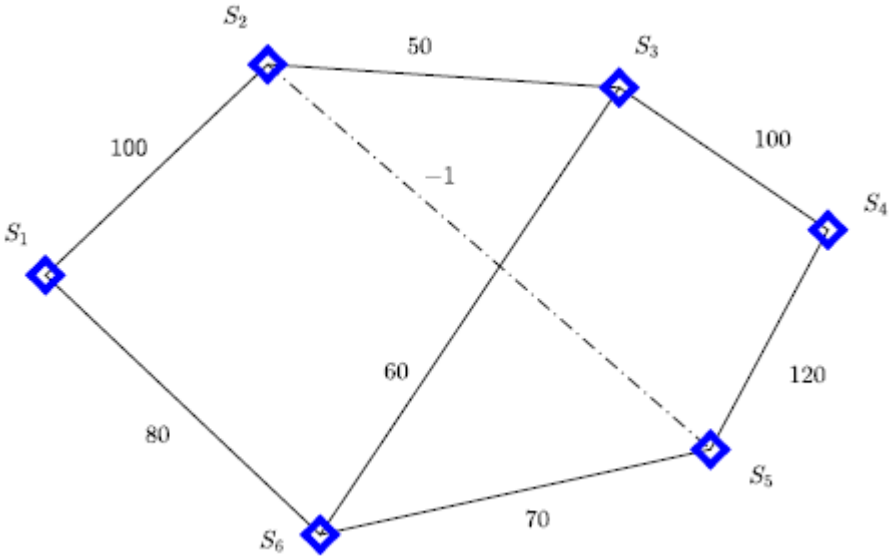
Example 1

Input
1 4
6 8
1 2 100
2 3 50
3 4 100
1 6 80
6,5 70
6 3 60
5 2 -1
5 4 120

Output
225

Explanation
One needs to travel from Star 1 to Star 4. There are 6 stars, and 8 paths listed. The first path listed says that travelling from 1 to 2 takes 100 units of time. The last path listed says that there is a wormhole between Star 5 and Star 2

A diagram of the star catalogue is given below.



It can be seen that if we do not use the worm hole, the shortest time taken is 1 - 6 - 3 - 4 of 240 units (1-2-3-4 takes more). If the wormhole is used, the path 1-6-5 gets us to 5 at 150. We can then reach 2 through the wormhole at T=75. Then 2-3-4 takes another 150, getting us to the destination at T=225.

Example 2

Input
1 4
5 8
1 2 10
1 5 20
5 2 20
2 3 70
5 3 50
3 4 20
5 4 100
4 2 -1

Output
90

Explanation
The start of the journey is Star 1, and the destination is 4. The shortest path from start to destination is 1-5-3-4, with a total time of 90. The wormhole starts at the destination, and if used, cannot end up at the destination earlier. The output is 90.