University of Computer Studies, Yangon

Asia-Yangon National Programming Contest

June 25, 2016

Problem A. Intelligent Square

Run Time Limit: 1 sec

A local traditional carpenter needs help in calculating the area of the different size of the square tables. Being the neighbor of the carpenter and also a current Computer Sciences student you would like to help him with a program which can calculate the area of the square when you increase or decrease the side of the square. And in order for him to be able to calculate easily your program needs to be able to calculate the area of the square with the increase and decrease of the side in percentage. Let's assume the side of the square as S and the area of the square as A. If S is increases by 40%, then A increases by how many %?

Input

The first line contains integer T(1<=T<=20) which is the number of test cases. Each test case has only one line and it contains an integer value which represents an increasing or decreasing percentage of the side of the square.

Output

For each test case, print out the case number followed by your best result possible containing percentage after an integer.

Output Format : Case_1:XX%

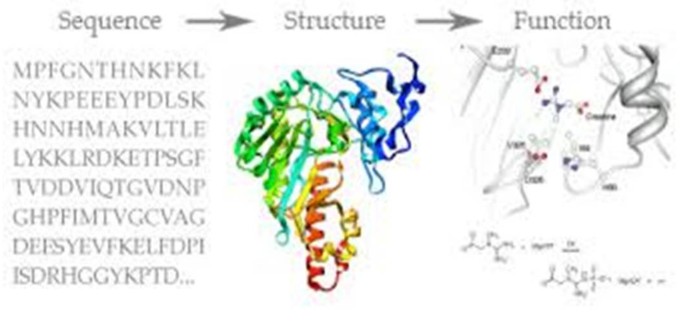
Case<underscore><case number><full colon><output value><percentage symbol>

Sample Input	Sample Output
4 20 80 65 -20	Case_1:44% Case_2:224% Case_3:172% Case_4:-36%

Problem B. Find Common Subsequence

Run Time Limit: 1 sec

In molecular biology, DNA information is represented as long genetic code sequences. Computers store genetic sequence information as simple rows of sequence characters called strings, which are similar to the sequences shown on the computer terminal.



In genetics research, as molecular biologists analyzed more and more genetic code sequences, they discovered that many organisms share similar genes that can be identified by their sequence similarity. When biologists find a new sequence, they typically want to know what other sequences it is most similar to and match. But exact matches rarely occur in biology because of small changes in DNA replication. For this reason, biologists need a method that computes similarities between sequences that do not match exactly.

One way of computing how similar two sequences is to find the maximum length of the common subsequences of two sequences. A subsequence is a sequence that can be derived from another sequence by deleting zero or more elements without changing the order of the remaining elements. For example, the sequences “ACT ” , “ATTC” , “T” **,** “ACTTGC” are subsequences of “ACTTGCG” obtained after removal of some elements in sequence. A common subsequence of two sequences is a subsequence that appears in both strings. Maximum length of common subsequence is the length (total number of characters in sequence) of longest common subsequence.

Example1 : Given two sequences “ACBDEGCEDBG” and “BEGCFEUBK” then some common subsequences of two sequences could be “BEE”, “BEEB”, “BEGCEB”. The longest common subsequence is “BEGCEB” and so maximum length of common subsequence is **6**.

Example2 : Given two sequences, CBGCAGDGCGd , GGCGACETdAEC, some common subsequences are “CC” , “AC” “ CCd ” , “GGG”, “GGCGd”, “CGCAC”, “GCGCd ” and in this case the maximum length of the common subsequences is **5**.

Example 3 : Given two sequences of DNA,

SEQ_1 = ACGGTGTCGTGCTATGCTGATGCTGACTTATATGCTA

SEQ_2 = CGTTCGGCTATCGTACGTTCTATTCTATGATTTCTAA

The two longest common subsequence of sequences SEQ_1 and SEQ_2 are : **CGTTCGGCTATGCTTCTACTTATTCTA** and **CGTTCGGCTATCGTGCTATTTATGCTA** and maximum length of the common subsequence is **27**.

Your task is to write a program that determines the maximum length of common subsequences of given two sequences. For computer analysis, each sequence consists of both uppercase and lowercase English letters and contains maximum 50 characters.

Input:

The first line contains an integer T (1<=T<=10) which is the number of test cases. The input for each test case has two lines. Each line contains one sequence.

Output:

For each test case, output an integer value which denotes the maximum length of common subsequence of the given two sequences.

Sample Input	Sample Output
4 CGTGCTATGCT GCTATCGTACGTTCTATTCTA TCttABc TtaCbbc abc def ThisIsProgContest AllStudentsTryHard	10 3 0 3

Problem C. Commercial and Industrial Zone

Run Time Limit: 1 sec

The authorities are planning and developing new commercial and industrial zone is not just about providing local jobs; it’s also about creating a supply chain of jobs that has a ripple effect in strengthening economic growth. The zone is sectoring by industrial groups. They decide to have different companies design each sector in new zone. All companies submit coordinates for their sectors and construction begins when the sectors do not overlap. Authorities hope this take a very long time to work out.

You have been hired to write a program that analyzes sector boundaries and report to authorities whether any sectors overlap. Two sectors are considered to overlap if one contains the part of the other, if they share the portion of the common side, or if they share the common corner.

Each company will submit the (x, y) coordinates of the lower left and top right corners of their sectors, where (0, 0) indicates the bottom left corner of the sector. You may assume that all sectors fit within a 100 by 100 area (coordinates range between 0…99).

Input

The input will consist of a line containing the number of sectors in new industrial zone, followed by one line per sector in the format: x1 y1 x2 y2 where (x1, y1) is the lower left-hand corner of the rectangle and (x2, y2) is the upper right-hand corner the rectangle.

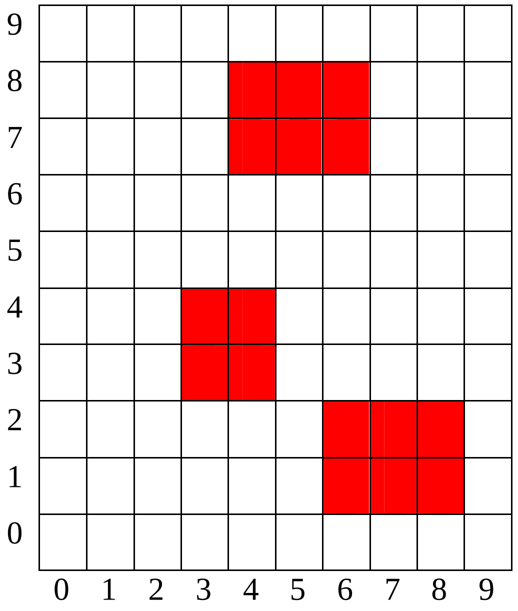
Output

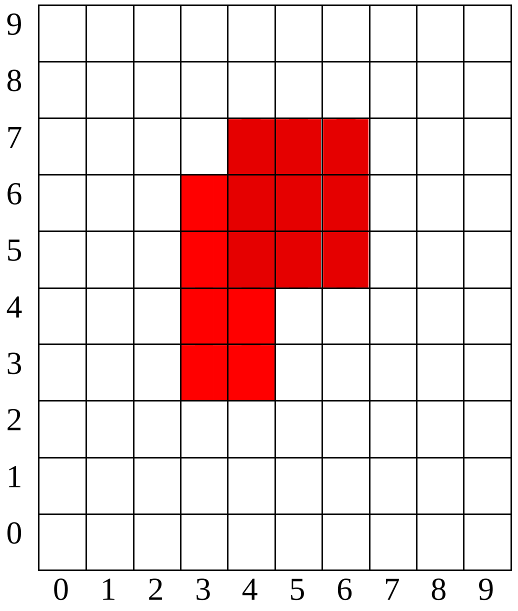
The output will be “OVERLAP” if any sector overlap, otherwise “NO OVERLAP”.

Sample Input	Sample Output

3
3 3 5 5
4 7 7 9
6 1 9 3
	NO OVERLAP

2
3 3 5 7
4 5 7 8
	OVERLAP





Problem D. Identical Digits

Run Time Limit: 1 sec

In the last ACM ICPC University final Contest, there were two teams as contestants. It is an interesting and exciting contest because the marks they got are shown on LED circuit board to all. At the start, their marks are shown as 0:0 for team A and team B respectively. They will get 1 mark for each problem. If team A solves a problem correctly and completely, the shown marks will be 1:0. If both teams cannot solve all the problems, it will be in 0:0. The marks are shown in digit integers. Therefore, there can be identical digit integer marks (or) non-identical digit marks. This means that:

identical digits are: 0:0, 1:1, 1:11, 2:2, 11:1, 2:22, 22:2, etc.

non-identical digits are: 0:1, 1:10, 10:10, 2: 20, etc.

One of the judges wants to know the followings while digit numbers are identical between two teams

(a) How many numbers of possible draw times?

(b) How many numbers of possible win times for team A?

(c) How many numbers of possible win times for team B?

Input

First line contains **T**, the number of test cases. The followings are the marks of both teams A and B for given test cases T.

Output

For each test case, three integers should be printed in one line. First is the number of possible draw times, the second is the number of possible win times of team A, and the last one is the number of possible win times of team B.

Constraints

- $1 \leq T \leq 10^2$
- $1 \leq$ marks ≤ 100

Sample Input	Sample output

6
38 60
23 37
10 12
11 8
1 1
4 11
	13 3 5
	12 2 3
	10 0 1
	9 1 0
	2 0 0
	5 0 1

Explanation for Test Case 1#

Input:

Team A marks = 38

Team B marks = 60

Output:

(a) The numbers of possible draw times while digits are identical= {0:0, 1:1, 2:2, 3:3, 4:4, 5:5, 6:6, 7:7, 8:8, 9:9, 11:11, 22:22, 33:33} = 13

(b) The numbers of possible win times of team A while digits are identical= {11:1, 22:2, 33:3} = 3

(c) The numbers of possible win times of team B while digits are identical= {1:11, 2:22, 3:33, 4:44, 5:55} = 5

Problem E. Notes Combination

Run Time Limit: 2 sec

Flora and Rita, her mother, are now in Yangon for visit. While they are staying and visiting in Myanmar, they always try some Myanmar foods in restaurants and go shopping to buy some things. One day, they went to the grocery store and Flora wants to buy some bread. She asks Rita for the 650 Kyats she needs. Rita asks her how she wants it as she didn't know combination. Flora considers how many ways there are to make 650 Kyats using three notes, 100Kyat, 500Kyat and 50Kyat. She realized that there are 9 ways to do this.

1. 650 = 100 + 100 + 100 + 100 + 100 + 100 + 50
2. 650 = 100 + 500 + 50
3. 650 = 100 + 100 + 100 + 100 + 100 + 50 + 50 + 50
4. 650 = 500 + 50 + 50 + 50
5. 650 = 100 + 100 + 100 + 100 + 50 + 50 + 50 + 50 + 50
6. 650 = 100 + 100 + 100 + 50 + 50 + 50 + 50 + 50 + 50 + 50
7. 650 = 100 + 100 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50
8. 650 = 100 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50
9. 650 = 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50 + 50

Flora realizes that if Rita comes alone next time to Myanmar, her mother would be in a trouble between the different currencies with different notes. So, Flora wants you to solve this problem by writing program to compute such combinations for her mother use. The problem is stated as follows:

Given a currency system (c_1, c_2, c_3, ...c_n), where c_i means a Myanmar note worth c_i Kyat, and a given number, , where K is the total Kyat, output the number of ways to obtain **K** Kyat using that currency system. Be ensuring that she just wants to know how many ways there are to combine notes, not what those ways are.

Input

The first line contains integer T(1<=T<=10) which is the number of test cases. The input for each test case is given in a single line. This line contains a number of integers: the first integer says the number of notes, ***n***, to use, following n different notes, following the total Kyat, **K**, to combine.

Output

For each line of input, produce three lines of output. The first line is the different Notes the system will use, the second line is the total Kyat to combine and the third line is the number of combinations. See the samples for the exact format of output.

Sample Input	Sample Output

2
3 100 500 50 650
4 200 50 500 100 570
	Notes=100,500,50
	Total=650
	Number of combinations=9
	
	Notes=200,50,500,100
	Total=570
	Number of combinations=0

Problem F. Secure Password Checker

Run Time Limit: 1 sec

Passwords are a traditional and widespread method of authentication, both on the Internet and off-line. Passwords are portable, easy to understand for laypersons, and easy to implement for the operator. Thus, password-based authentication is likely to stay for the foreseeable future. However, users tend to choose weak passwords, thus these passwords can easily be guessed. To ensure an acceptable level of security of user-chosen passwords, sites often use mechanisms to test the strength of a password and then reject weak passwords.

The table below summarizes one set of criteria for determining the strength of a password, and shows how to apply these criteria to two different sample passwords.

Sample 1:UCSY*(123comp4567)

Sample 2: 9876598765

Category	Description	Sample 1 (Score)	Sample 2 (Score)	Remark
Length	Score 4 points for each character in the password.	+72	+40	Addition Case
Basic Requirements	To qualify, must be at least 8 chars long and also contain 3 of the four basic character types (upper case letters, lower case letters, digits, and symbols*). Score two points for the length plus another two for each of the four types of characters it contains.	+10	--	Addition Case
Upper Case	Add (length – n) * 2, where n is the number of upper case letters and n > 0	+28	--	Addition Case
Lower Case	Add (length – n) * 2, where n is the number of lower case letters and n > 0	+28	--	Addition Case
Digit	Add 4 * n, where n is the number of digits, but only if n < length.	+28	--	Addition Case

Symbols	Add 6 * n, where n is the number of symbol characters	+18	--	Addition Case
Letters Only	If the password contains only letters, subtract 2 point for each letter.	--	--	Subtraction Case
Digits Only	If the password contains only digits, subtract 2 point for each digit.	--	-20	Subtraction Case
Total Score	Add up all the above. Negative scores become 0. Scores over 100 become 100. 0-20 = Very Weak, 21-40 = Weak, 41-60 = Good, 61-80 = Strong, 81-100 = Very Strong	100 Very Strong	20 Very Weak	

*A symbol is any character that is not a letter or digit.

The problem is stated as follows: Given a password, you will have to find out the strength of that password.

Input

The first line of the input contains the number of test cases, T (1≤T≤20). Then T test cases follow in next T lines. Each line contains a string to check the strength of it.

Output

For each test case produce one line of output which contains strength level of the password and its total score.

Example

Sample Input	Sample Output

4
ACPC$%987zyxw4321
9876598765
UCSY()123
abbbbbb1
	Case_1:Very Strong(score=100)
	Case_2:Very Weak(score=20)
	Case_3:Strong(score=78)
	Case_4:Weak(score=38)

Problem G. Coloring Polygon

Run Time Limit: 1 sec

The students in a classroom are given an assignment to fill the beautiful colors to the polygon shape. First group has to design to form the symmetric polygons with n vertices. Third group fills colors to that polygon. Before filling, second group is responsible to count the number of intersection at its vertices. Polygons are filled colors from bottom to top one scan line by line, a dashed line as shown in figure. In that case, it is important not to fill the outside part of the polygon. To fill inside the boundary of the polygon, it needs to know how many numbers of intersections at each vertex along the scan line.

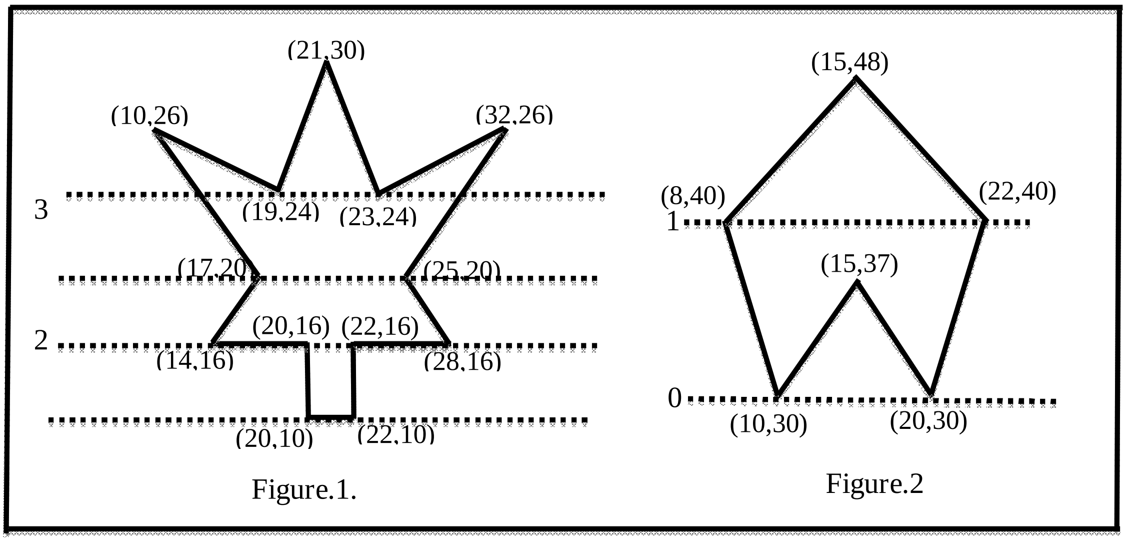


Figure.1.

Figure.2

Input

First line contains **T**, the number of test cases. The second line shows an integer n, the number of vertices of the polygon in clockwise order. This is followed by n lines, each containing the two integers x and y coordinates for a polygon vertices. The vertices coordinate are sequentially given from the top vertex.

Output

For each case, print the required answer in one line with separated integers, which describe the total numbers of intersections on all the vertices at each scan line from bottom to top.

Constraints

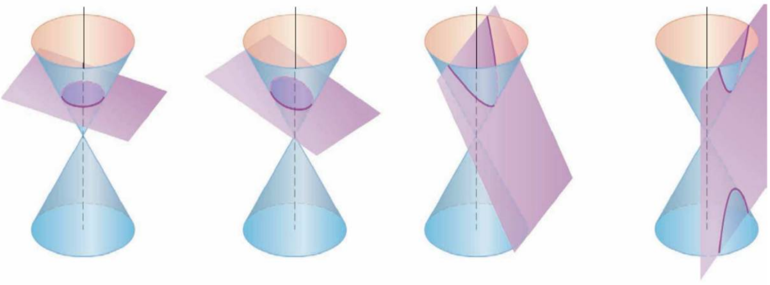
- $1 \leq \ T \leq 10^5$
- $3 \leq n \leq 30$
- $1 \leq \ x, y \leq 10^9$
- Polygons are vertically symmetric and the edges are no self-intersection.
- Scan lines and intersections are only considered at the vertices which have upper vertices.
- Start vertex is also End Vertex in clockwise order.

Sample Input	Sample output
2	2 2 2 4
14	4 2
21 30	
23 24	
32 26	
25 20	
28 16	
22 16	
22 10	
20 10	
20 16	
14 16	
17 20	
10 26	
19 24	
21 30	
7	
15 48	
22 40	
20 30	
15 37	
10 30	
8 40	
15 48	

Problem H. Conics

Run Time Limit: 2 sec

Conics or *conic sections* (circle, ellipse, parabola and hyperbola) are formed by cutting a double cone with a plane as shown in figure. The **general form** of the equation for each conic is: $\boldsymbol{Ax^2 + Bxy + Cy^2 + Dx + Ey + F = 0}$. They differ according to the coefficients. For the circle, A and C must be equal and B must be 0. An example of an equation of a circle is: $x^2 + y^2 + 2x - 6y + 6 = 0$. For the ellipse, A and C cannot be equal but must have the same sign and B must equal 0. An example of an equation of an ellipse is: $x^2 + 4y^2 - 6x - 16y - 11 = 0$. For a parabola, A or C must equal 0 and B must be 0. An example of an equation of a parabola is: $x^2 + 6x + 16y + 11 = 0$. For a hyperbola, A and C must have different signs and B must equal 0. An example of an equation of a hyperbola is: $4x^2 - y^2 + 8x + 4y + 11 = 0$.



Circle: plane perpendicular to cone axis

Ellipse: plane oblique to cone axis

Parabola: plane parallel to side of cone

Hyperbola: plane cuts both halves of cone

To obtain more information (center, major axis, etc.,) about conics, the equation must be modified. The method commonly used is called **“completing the square”**. Using the equation of the circle above, the result is as follows:

$$x^2 + y^2 + 2x - 6y + 6 = 0$$

$$(x^2 + 2x + \mathbf{1}) + (y^2 - 6y + \mathbf{9}) = -6 + \mathbf{1} + \mathbf{9}$$

$$(x + 1)^2 + (y - 3)^2 = 4$$

From this form, the center of the circle is at (-1, 3). Finding information about the ellipse is done in a similar manner.

$$x^2 + 4y^2 - 6x - 16y - 11 = 0$$

$$x^2 - 6x + \mathbf{9} + \mathbf{4}(y^2 - 4y + \mathbf{4}) = 11 + \mathbf{9} + \mathbf{16}$$

$$(x-3)^2 + 4(y-2)^2 = 36$$

$$\frac{(x-3)^2}{36} + \frac{(y-2)^2}{9} = 1$$

From this form, the center of the ellipse can be found to be at (3, 2) and the major axis is x=3.

Using the procedure above, examples of equations of a hyperbola in standard form are as follows:

$\frac{(x-3)^2}{36} - \frac{(y-2)^2}{9} = 1$, center at (3, 2) and the principal axis is the line y=2.

$\frac{(y-2)^2}{36} - \frac{(x-3)^2}{9} = 1$, center at (3, 2) and the principal axis is the line x=3.

Again, using the procedures above, examples of equations of a parabola in standard form are as follows:

$(x-3)^2 = 4(y-1)$, $\qquad\qquad$ $(y-1)^2 = 4(x-3)$

The vertex of both parabolas is (3, 1). The axis of symmetry for the left parabola is x=3 and for the right, the axis of symmetry is y=1.

Input

The first line contains an integer, T(1<=T<=20), which is the number of test cases. Each test case consists of a string representing an equation in general form. The ^ will be used to denote exponents.

Output

For each set of data, print the type of shape according to the input equation. If it is a circle, print its center in (x, y) format. If it is an ellipse, print its center and its major axis. If it is a hyperbola, print its center and the equation of the principal axis. If it is a parabola, print its vertex and the equation of the axis of symmetry. Otherwise print "Others".

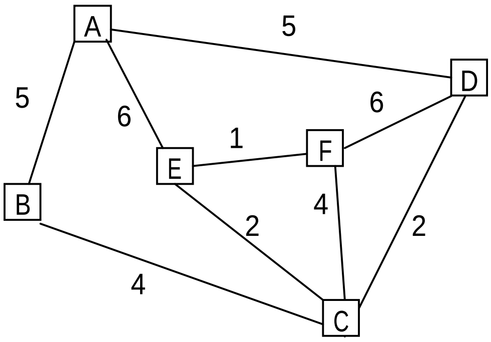
Sample Input	Sample Output
5 x^2+4y^2-6x-16y-11=0 x^2+y^2+2x-6y+6=0 -4x^2+y^2+24x-4y-68=0 x^2-6x-4y+13=0 y^2-4x-2y+13=0	Case_1:Ellipse,(3.0, 2.0),x=3.0 Case_2:Circle,(-1.0, 3.0) Case_3:Hyperbola,(3.0, 2.0),x=3.0 Case_4:Parabola,(3.0, 1.0),x=3.0 Case_5:Parabola,(3.0, 1.0),y=1.0

Problem I. Shortest Distance between Towns

Run Time Limit: 1 sec

In this problem, a salesperson wishes to travel between two towns. Towns are joined together by roads. Roads have a certain distance. Not all towns are connected directly to the other towns. It may be necessary to travel through other towns to get to a desired town. The salesperson wishes to travel the shortest distance between the two towns.

A map of the towns can be depicted as follows:



Input

The input will contain information about the map of the towns and then information about the towns the salesperson wishes to travel to. The first line of the input contains ***N***, the number of roads on the map, ***1 <= N <= 30***. The next ***N*** lines will contain information about the roads, the starting town letter ***(A - M)***, the ending town letter ***(A - M)***, a space and then ***D***, an integer, the distance between these towns, ***1 <= D <= 50***. The ending town letter will never be the same as the starting town letter and the road to be bidirectional. It is possible that many roads might exist between a pair of city. After these lines there will be K lines of data about the towns the salesperson wishes to travel to. Each of these K lines contains two letters, the starting town and the ending town.

Output

The output K lines contain the towns list between the starting town and the ending town and the shortest distance where the salesperson wishes to travel to.

Sample Input	**Sample Output**
9	
AB 5	ADC,7
AD 5	FECB,7
EA 6	BCD,6
BC 4	ECD,4
CD 2	AEF,7
EC 2	
CF 4	
DF 6	
EF 1	
AC	
FB	
BD	
ED	
AF	

Problem J. Zoo Story

Run Time Limit: 1 sec

In Yangon Zoo, N locked cages have birds inside all sleeping. A “City Tour Trip” including N contestants (University Students) from ACM-ICPC Asia Yangon National Programming Contest 2016 comes to the Zoo. When they arrive at Zoo, first student goes to all cages and opens all. Second student goes to cage 2, cage 4, cage 6, etc. and opens locked cages, and locks opened cages. Third student goes to cage 3, cage 6, cage 9, etc. and opens locked cages, and locks opened cages. The problem is how many cages are opened after all N students do the above actions.

Input

The first line contains integer T (1<=T<=20) which is the number of test cases. The input for each test case is given in a single line. This line contains one integer **NUMBER**: here **NUMBER** means the total number of cages and also the total number of students in the Zoo.

Output

For each line of input, produce several line of output. Each output line contains the information which cage number is opened. See the samples for the exact format of output.

Sample Input	Sample Output
3 10 20 30	Cagenum 1 Cagenum 4 Cagenum 9 Cagenum 1 Cagenum 4 Cagenum 9 Cagenum 16 Cagenum 1 Cagenum 4 Cagenum 9 Cagenum 16 Cagenum 25