Problems Overview

Problem A: Stock Market

Problem B: Sum

Problem C: ATM withdrawal

Problem D: Treasure Box

Problem E: ACM

Problem F: Coupled Polygons

Problem G: Production Planning

Problem H: Pencil Game

Problem I: Space Tour

Problem J: Math Magic

Note: The input and output for all the problems are standard input and output.

Problem A: Stock Market

An expert in deep learning has developed a new method for stock market forecasting. According to his method, the forecasted stock prices of VAIP for the next N days are $p_1, p_2, p_3, \ldots, p_N$, where p_i denotes the forecasted stock price for the i^{th} day.

Given p_i and the total amount of money W, find a way to buy and sell VAIP stocks to maximize the profit. You can only buy VAIP stocks in one day, or you don't buy at all. The number of stocks bought or sold must be an integer.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 10. The following lines describe the datasets.

Each dataset consists of two lines where the first line contains two space-separated integers N ($N \leq 100$) and W ($0 < W \leq 1{,}000{,}000$) where W is the amount of money you will invest in the stock. The second line contains N space-separated integers representing p_i ($0 < p_i \leq 1000$).

Output

For each dataset, write out on one line the maximum profit you could make.

Sample Input	Sample Output
1 12 1000000 99 100 101 102 100 99 97 101 102 105 104 104	82472

Explanation:

You should buy 10309 (=1000000 div 97) stocks at the price of 97 and sell 10309 stocks at the price of 105 to get the maximum profit of (105-97)*10309

Problem B: Sum

Write a program to compute the following sum S given a positive integer n:

$S = \left\lfloor \frac{n}{1} \right\rfloor + \left\lfloor \frac{n}{2} \right\rfloor + \cdots + \left\lfloor \frac{n}{n} \right\rfloor$, where $\lfloor x \rfloor$ is the largest integer not greater than x.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 30. The following lines describe the datasets.

Each dataset contains a positive integer n ($n \leq 10^{12}$) written on a separate line.

Output

For each dataset, write in one line the remainder of the computed sum S divided by 10^6.

Sample Input	Sample Output
2 1 5	1 10

Problem C: ATM withdrawal

Vinh works for an ATM machine manufacturing company. The basic functionality of an ATM machine is cash withdrawal. When a user requests a cash withdrawal of W VND (Vietnamese Dong), the ATM has to dispense N money notes such that they sum up to W. For the next generation of ATM machine, Vinh is working on an algorithm to minimize the number N of money notes for each cash withdrawal transaction.

Your task is to help Vinh to do his job given that the money notes come in the values of 1000, 2000, 3000, 5000, 1000×10^1, 2000×10^1, 3000×10^1, 5000×10^1,..., 1000×10^c, 2000×10^c, 3000×10^c, 5000×10^c where c is a positive integer and Vinh has unlimited supply of money notes for each value.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 1000. The following lines describe the datasets.

- The first line consists of one positive integer W ($W \le 10^{18}$);
- The second line consists of one positive integer c ($c \le 15$).

Output

For each dataset, write in one line two space-separated integers N and S where S is the number of ways to dispense the fewest number N of money notes. In case there is no way to serve the cash withdrawal request, write out 0 in one line instead.

Sample Input	Sample Output
2 1000 1 7000 1	1 1 2 1

Problem D: Treasure Box

Your team was exploring an ancient city. Suddenly you found an old scroll with 2 integer numbers N and K, which encrypts the secret code to open a treasure box. Considering a transformation on an integer X described as follows:

$$X = X + X \bmod 100,$$

the secret code can be obtained by applying the above-described transformation K times successively to N.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 500.

Each dataset has two space-separated positive integers N and K ($1 \le N \le 10^9$, $1 \le K \le 10^9$) written on a single line.

Output

For each dataset, write on a single line the secret number decrypted from N and K.

Sample Input	Sample Output
2 31102014 2 10101 10	31102056 10324

Problem E: ACM

An IT company puts out an advertisement to hire staff for a newly established office. They received n applications and organized an interview to select the best ones. They want to recruit candidates with both high level of expertise and good teamwork skill. Each candidate is assigned an ACM (**A**bility **C**oefficient of **M**ulti-collectives) score that represents how the candidate meets the company's internal selection criteria. Initially, everyone's ACM score is *1*.

Candidates are arranged into a round table of n seats, indexed from *1* to n. The first person sits next to the second person and the n^{th} person. For each interview question, candidates with indices from L and R form a group and register their collective answer to the system. If $L \le R$, the group consists of candidates at indices L, L+1, L+2, . . ., R. If $L > R$, the group consists of candidates at indices L, L+1, . . ., N, 1, . . ., R. Depending on the answer, the *ACM* score of each group member is either multipled by X or divided by Y (in the later case, it is guaranteed that all ACM scores of the group are divisible by Y).

During the interview, the company may also request the system to output the product of the *ACM* scores of a group. This product could be a large number, so the system has to only output the value at modulo P.

In summary, the system has to handle the following three types of queries:

- *0 L R P* – compute the product of the ACM scores of all candidates from L to R, modulo P
- *1 L R X* – the ACM score of each candidate from L to R is multiplied by X
- *2 L R Y* – the ACM score of each candidate from L to R is divided by Y

For every query, we have $1 \le L, R \le N$, $1 \le P \le 10^9+7$, $1 \le X, Y \le 150$.

Your task is to implement the system and output the computed products for every query of type ***0***.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

Each dataset comes in the following format:

- The first line contains 2 integers n, m where n is the number of candidates and m is the number of queries to be processed ($1 \le n, m \le 50000$).
- In the next m lines, the i^{th} line contains the i^{th} query.

Output

For each dataset, write out the corresponding outputs for queries of type ***0*** where each query output is on a separate line.

Sample Input	Sample Output
2 6 5 0 1 5 1000000007 1 2 4 15 0 1 6 8704173 2 2 3 3 0 1 6 1000000007 6 6 1 1 4 20 1 2 6 15 0 1 6 9704331 2 3 6 5 0 1 4 1000000007 0 1 5 1000000007	1 3375 375 1775880 21600000 64800000

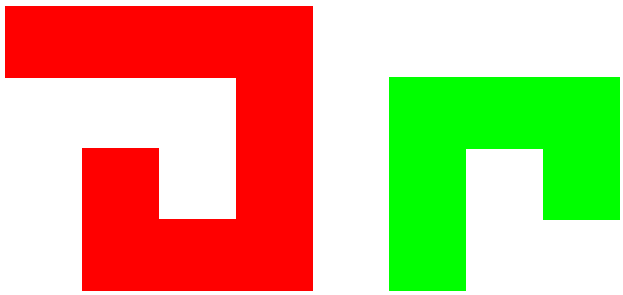
Problem F: Coupled Polygons

A nice polygon is defined as:

- A non-self-intersecting polygon;
- All edges are parallel to the horizontal or vertical axes;
- All edges have integer unit of measure length.

A nice polygon can be encoded by a string consisting of only four type of characters: 'N', 'E', 'W' and 'S' standing for North, East, West and South, respectively. The polygon can be reconstructed from the string by drawing the edge following the direction described, with each character represents exactly 1 unit long.

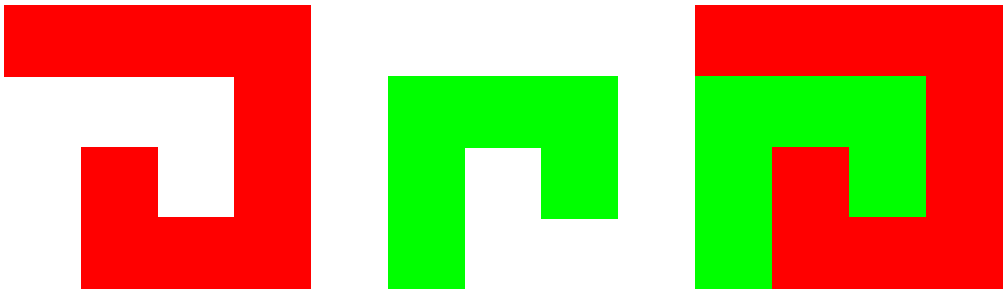
For example, the following 2 polygons are encoded as EEEESSSSWWWNNESENNWWWN and NNNEEESSWNWSSW, respectively.



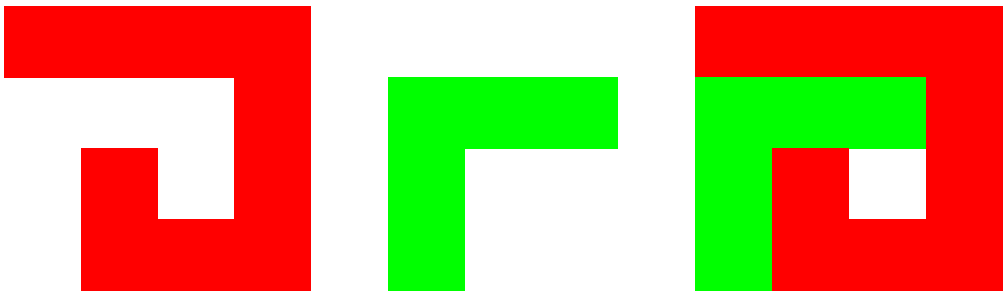
A nice polygon B is said to be a perfect match for a nice polygon A if:

- Only by moving the polygons horizontally or vertically, these two polygons can be positioned so that they do not overlap each other and form a square with no holes inside (i.e., the area of the formed square is equal to the sum of the two polygons' areas)
- The area of B is minimum.

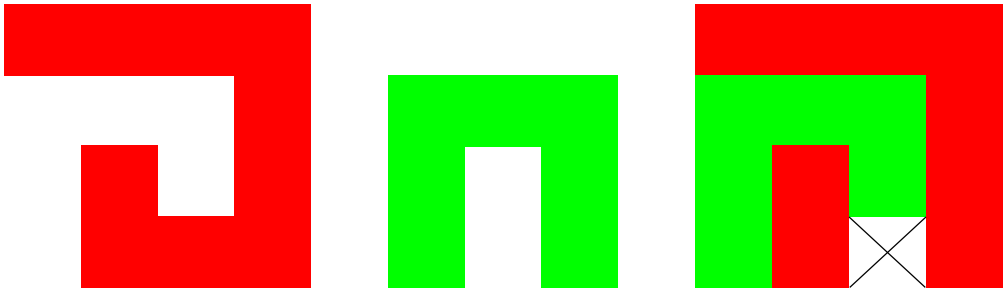
For example, the light-colored polygon NNNEEESSWNWSSW is a perfect match for the dark-colored polygon EEEESSSSWWWNNESENNWWWN, illustrated as follows:



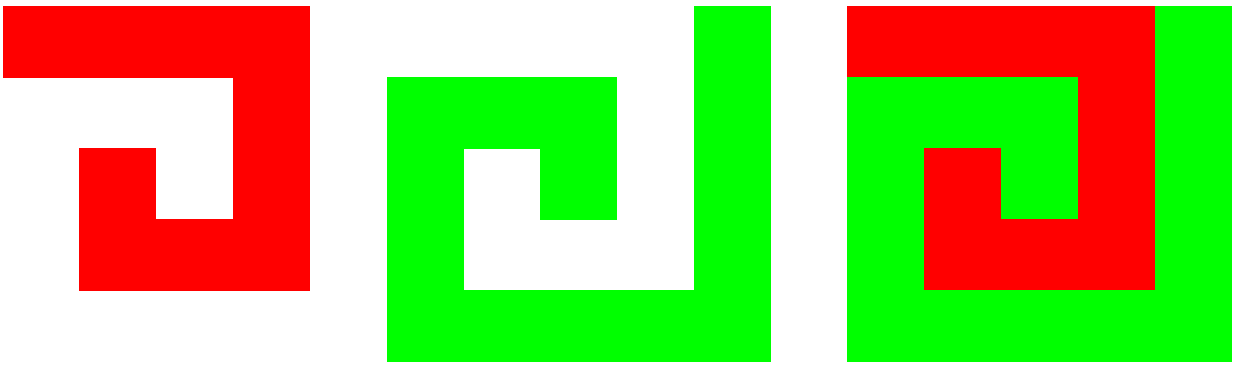
In contrary, the following two polygons (light and dark) cannot form a square with no holes inside, hence they cannot be a perfect match for each other.



The following two polygons are not a perfect match for each other either, as they cannot form a square without overlapping each other



In the following example, the two polygons form a perfect square but the light-colored polygon is not a perfect match for the dark-colored polygon, as the light-colored polygon is not the smallest one to form a square. The dark-colored polygon is, however, a perfect match for the light-colored polygon.



Given a nice polygon A, your task is to find a nice polygon B such that B is a perfect match for A.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 30. The following lines describe the datasets.

Each dataset has one string representing the polygon A on a single line. The string consists of only 'N', 'E', 'W' and 'S' and contains at most 100,000 characters.

Output

For each dataset, write on one line a string representing the nice polygon B that is a perfect match for A. If there are multiple answers, write out the one that come first lexicographically. If there is no solution, write out the string "NONE" on one line.

Sample Input	Sample Output
1 WSSEEENNNNWWWWSEEESSWN	EEESSWNWSSWNNN

Problem G: Production Planning

LazyCorp is preparing its production plan for a big interplanetary sporting event to be held in planet Alpha. LazyCorp is capable of manufacturing n types of products that require n-1 types of materials. Anticipating the huge demand from the sporting event, LazyCorp has stocked the required materials for production. Currently, the company has b_i material of type i (i = 1, 2,…, n−1). Based on previous years statistics, LazyCorp has identified the consumption rate of each material type for production. The consumption rate is captured by a coefficient matrix $A = (a_{ij}: i = 1, 2,\ldots, n-1; j = 1, 2, \ldots, n)$, where a_{ij} is the amount of type i material needed to manufacture one product of type j. We can assume that the rank of A is n-1. Based on market study and forecast, LazyCorp has also figured out the expected profit from one unit of type j product to be c_j, j = 1, 2, …, n. A production plan of the company can be represented by a vector $(x_1, x_2,\ldots, x_n)$ where x_j is a non-negative integer indicating the quantity of type j product to be manufactured, j = 1, 2, …, n.

LazyCorp board of directors is interested in working out if they can have a production plan that use up all n−1 types of materials in stock (i.e., the amount of type i material to be used in the production plan has to be exactly b_i, i = 1, 2,…, n−1). In case there exists such a production plan, they are interested in identifying the production plan that would bring the maximum expected profit.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

- The first line contains an integer n ($n \le 200$), which is the number of product types.
- The second line contains n positive integers $c_1, c_2,\ldots, c_n$ ($c_j \le 1000, j = 1, 2, \ldots, n$);
- The third line contains n−1 positive integers $b_1, b_2,\ldots, b_{n-1}$ ($b_i \le 10^6$, i = 1, 2, …, n−1);
- The u^{th} line in the next n−1 line contains n positive integers, each not exceeding 10^6, a_{u1}, a_{u2}, …, a_{un}; u = 1, 2, …, n−1.

Two consecutive numbers on the same line are separated by a space.

Output

For each dataset, write on a single line an integer that is the expected profit of the found production plan or -1 in case there is no such production plan.

Sample Input	Sample Output
2 3 1 2 3 20 100 1 1 1 2 3 5 2 1 5 100 3 12	60 -1

Problem H: Pencil Game

Minh has a box of pencils. The box is a rectangle of size $M \times N$, where position (i, j) has a pencil with a length of exactly $i \times N + j$ ($0 \le i \le M$-1, $0 \le j \le N$-1). Note that position (0, 0) does not have any pencil hence having a length of 0.

He wonders if he could select a sub-rectangle of the box and join all the pencils within that sub-rectangle together, to get a new long pencil that has a specific length L that he wants.

Your task is to find a sub-rectangle of the box in which the total length of the contained pencils is L and return the area of that the sub-rectangle. If there are multiple solutions, return the smallest possible area. If there's no such sub-rectangle, return -1.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 150. The following lines describe the datasets.

Each dataset contains three space-separated numbers M, N and L ($1 \le M, N \le 10^6$, $1 \le L \le 10^{12}$) written in one line.

Output

For each dataset, write in one line the smallest possible area of the sub-rectangle in which the total sum of pencil lengths is L. Write in one line -1 if there is no such sub-rectangle.

Sample Input	Sample Output
4 2 3 8 3 3 10 3 3 36 1000000 1000000 1000000000000	4 -1 9 2

Problem I: Space Tour

Alpha Centauri-M (ACM) is a planet with marvelous scenes for visitors from the Earth. Luckily you win a ticket to participate in a space tour to Alpha Centauri-M.

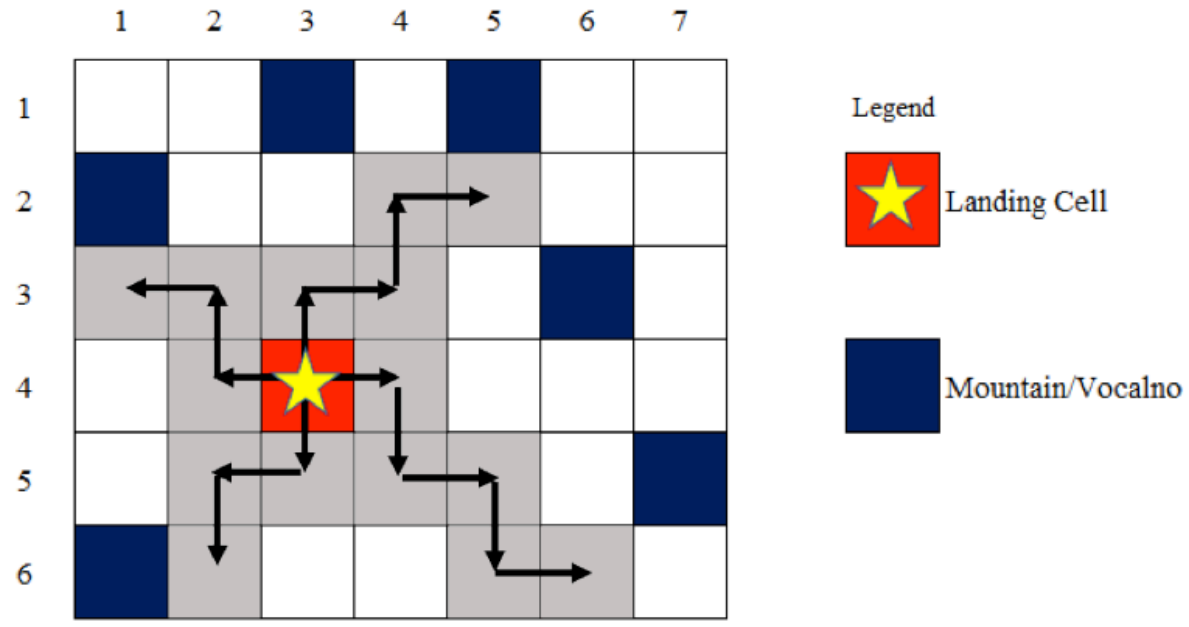
The map of the planet ACM is divided into a grid of M rows and N columns. The columns are indexed from 1 to N (from left to right) and the rows are indexed from 1 to M (from top to bottom). There are several cells on the map that are not safe for visitors to explore because of high mountains and volcanoes.

Upon arriving at ACM, the spaceship will land on any safe cell on the grid and passengers can visit the planet via a special system of space cars. From the landing cell (r_0, c_0) on the map, a space car can go to either one of the four connected adjacent cells, namely $(r_0 - 1, c_0)$, $(r_0 + 1, c_0)$, $(r_0, c_0 - 1)$, $(r_0, c_0 + 1)$. Subsequently, the space car will continue moving the following fixed navigation pattern:

1. Turn right and go forward one cell
2. Turn left and go forward one cell
3. Go back to step 1.

A space car can only visit safe cells, therefore it will stop if the next cell is not safe or is beyond the map boundaries.

The following figure illustrates a map consisting of $M = 6$ rows and $N = 7$ columns. From the landing cell (4, 3), you may visit 16 cells (including the landing cell).



For each landing cell on the map, you can determine the number of cells that you can visit (including the landing cell). Your task is to choose the landing cell from which you can visit the maximum number of cells on Alpha Centauri-M.

Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 20. The following lines describe the datasets.

The first line of a dataset contains 2 space-separated positive integers M and N ($1 \le M, N \le 1000$). The i^{th} line in the next M lines of the dataset contains N binary digits to represent the states of all cells in the i^{th} row of the map (1: safe cell, 0: unsafe cell).

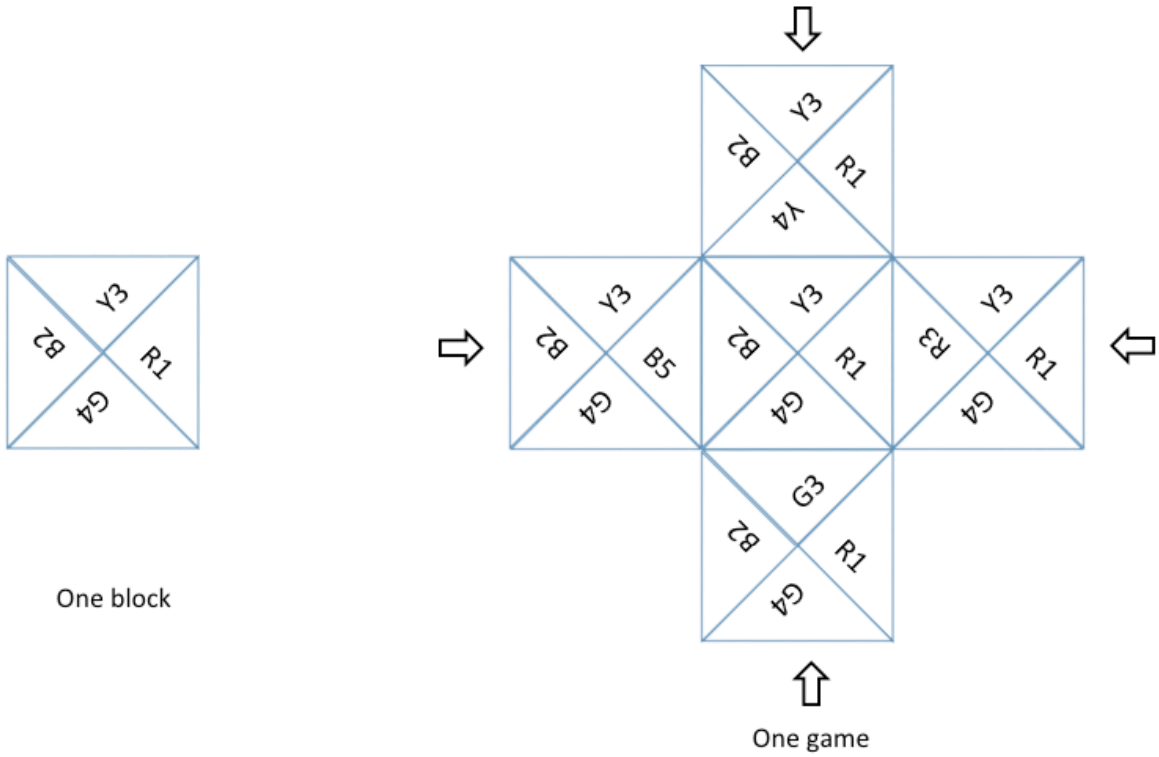
Output

For each dataset, write in one line the maximum number of cells that you can visit on Alpha Centauri-*M* (including the landing cell).

Sample Input	Sample Output
2 3 3 011 111 111 6 7 1101011 0111111 1111101 1111111 1111110 0111111	8 20

Problem J: Math Magic

A Math Magic game consists of a number of equal-sized blocks. Each block in Math Magic is divided into 4 triangles where each triangle takes a color from Blue (B), Green (G), Red (R), or Yellow (Y) and an integer from 0 to 9.

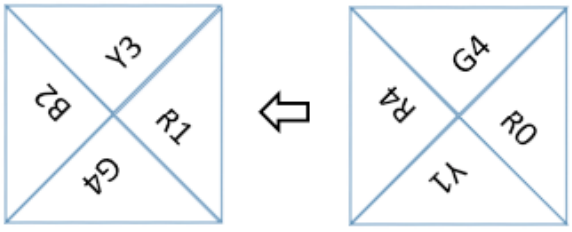
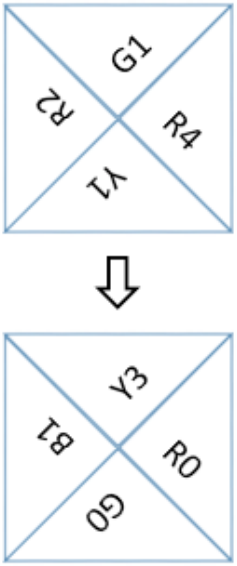


Starting with the center block, at each step a player can adjoin a new block in one of four directions along the horizontal or vertical axes centered at the center block. The new block is attached to an existing block in such a way that the two corresponding adjoined triangles must have the same color. The score of each block is calculated based on the color of the adjoined triangle, the integer in the adjoined triangle of the new block and the sum of four integers in the new block as follows:

Blue	Green	Red	Yellow
$S - x$	$S + x$	$S * x$	S % x if x > 0. Otherwise 0. %: Integer division, e.g., 10 / 3 = 3

If a block cannot be added in the board or the player does not wish to include in the board, then its score is $-S$ and the block is skipped. Notice that the center block, i.e. the first block, does not have a score.

For example, consider the following steps to add a new block and their corresponding scores

	$S = 4 + 4 + 0 + 1 = 9$ $x = 4$ Color of the adjoined triangle: red Score = $S * x = 9*4 = 36$
	$S = 1 + 3 = 4$ $x = 3$ Color of the adjoined triangle: yellow Score = 4 % 3 = 1

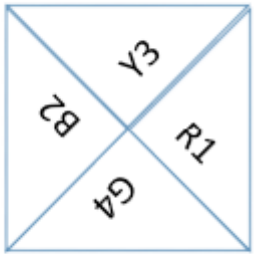
Given a list of blocks, a player will use the first one as the center block and either add or skip the remaining blocks to the board one at a time in sequence. The blocks can be rotated before being added to the board but they must be considered in the same order as given.

Your task is to find the maximum total score a player could get.

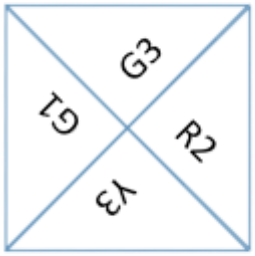
Input

The input file consists of several datasets. The first line of the input file contains the number of datasets which is a positive integer and is not greater than 40. The following lines describe the datasets.

The first line of each dataset contains an integer n ($n \leq 250{,}000$), which is the total number of blocks including the center block as the first one. For the next n lines, each line contains 8 characters representing the colors and integers for the block's triangle in clockwise order: $c_1n_1c_2n_2c_3n_3c_4n_4$ where c_i (c_i takes values from 'Y', 'R', 'G', or 'B' standing for yellow, red, green or blue respectively), n_i (n_i takes an integer value from 0 to 9) is the color and integer of the i triangle respectively ($1 \leq i \leq 4$).



Y3R1G4B2



G3R2Y3G1

Output

For each dataset, write on one line the maximum point score a player could get.

Sample Input	Sample Output
1 10 G3B7B3Y6 R9B2B0G3 G0Y2Y1R7 G2B2B7G9 G2B9Y3G1 R9B1R4Y8 G5G3Y1R6 G2R6R5B4 R6R6R3B4 R2B4Y4G3	517