Problems Overview V6.1

Problem A: Social Survey

Problem B: Rectangles

Problem C: Bookshelves

Problem D: Flash

Problem E: Hanoi Tower

Problem F: Protein secondary structure prediction

Problem G: Urbanization

Problem H: Digits

Problem I: Magic Sequence

Problem J: Parentheses

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

Problem A: Social Survey

Sociologists are interested in conducting a survey on social behavior of residents living on the same street clustered by income. They pick n households of a particular street where each house's street number ranging from 1 to n is considered its unique identifier. Each house's income is calculated by summing up the monthly incomes in Vietnam *dong* of all residents living in that house. They want to study samples of neighboring houses with similar income. Specifically, a t-sample of size k consists of k neighboring houses having consecutive identifiers from i to $i+k-1$ where any two houses' incomes in this sample differ by at most t *dongs* ($|income_p - income_q| \le t$ where $p,q \in [i,..,i+k-1]$ and $income_p$ is the income of house p).

Given the incomes of n houses and the threshold t, your task is to help the sociologists to find the largest t-sample.

Input

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

Each data set consists of two lines where the first line contains two integers n $(1 \le n \le 3\times10^6)$ and t $(0 \le t \le 3\times10^9)$ separated by space. The second line contains n integers (in the range from 1 to 5×10^9) separated by space representing the incomes of n houses in order.

Output

For each data set, write the size of the largest t-sample.

Sample Input	Sample Output
2 8 3 8 1 3 5 6 8 6 1 10 4 6 7 7 9 11 9 9 10 8 8	4 9

Problem B: Rectangles

Given the following 4 sequences of n positive integers:

A: $a_1, a_2, \ldots, a_n$

B: $b_1, b_2, \ldots, b_n$

C: $c_1, c_2, \ldots, c_n$

D: $d_1, d_2, \ldots, d_n$

Your task is to count the number of quadruples (i, j, k, t) such that the four integers a_i, b_j, c_k, d_t correspond to the sizes of a rectangle.

Input

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

Each data set consists of five lines where the first line contains the number of positive integers n $(n \leq 1000)$. Each of the following 4 lines contains one sequence of n space-separated positive integers not exceeding 10^9.

Output

For each data set, write the number of satisfied quadruples.

Input	Output
2 2 1 2 3 4 4 3 1 2 2 1 2 2 1 3 4 4 3	4 4

Problem C: Bookshelves

A library decides buy bookshelves to store its books and they need to determine the bookshelves' heights and widths in order to minimize storage costs. The books in the library have n different heights: h_1, h_2, ...,h_n. Let l_i be the number of books of height h_i (i = 1, 2, ...,n) in the library. Let's suppose that all the books have the same thickness of 1 centimeter and have the same width.

The bookshelves can have different heights and widths but they have the same depth which is the width of the book. Ignoring the book heights, a bookshelf to store k books need to have the width of at least k centimeter as each book is *1* centimeter thick.

A bookshelf of height h_i can be used to store books of height h_i or smaller. The cost for a bookshelf with width x and height h_i is $f_i + c_i x$, where f_i is a height-dependent cost which is independent of the width, and c_i is the height-dependent cost per one centimeter width.

Your task is to write a program to determine the minimum cost of shelving all the books.

Input

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

Each data set consists of five lines. The first line contains the number of book heights, n ($2 \le n \le 10000$). On the second line, there are exactly n integers h_i (i = 1,...,n) separated by spaces. On the third line, there are exactly n integers l_i (i = 1,..., n) separated by spaces. On the fourth line, there are exactly n integers f_i (i = 1,..., n) separated by spaces. The last line consists of exactly n integers c_i (i = 1,..., n) separated by spaces.

Output

For each data set, write in one line the minimum total cost of shelving all the books.

Sample Input	Sample Output
1 3 3 2 1 1 1 1 55 10 15 5 60 60	70

Problem D: Flash

Computer game addiction has become a disease that requires special medical treatment. The Central Neurological Hospital is testing a promising new method. In the middle of a computer game, they periodically pop a window up in the center of the screen containing ***m*** bright columns, each column displays a color in the range from 1 to 255 (256 color mode not including black). The window appears just seconds, which means it is not long enough for the eye to store the image. The process does not affect game playing but the brain of the player can still record information unconsciously.

If suitable sequences of colors are selected the players will get tired soon or feel uncomfortable, gradually leading to be bored or afraid of computer games. Given a sequence S of infinite number of color columns, consecutive windows of ***m*** columns will be selected from S to present to users in order. To generate S, the color columns are selected as follows. Starting with an initial set of ***n*** columns where column ***i*** has color c_i ($0 < c_i \le 255$, $i = 1,2,\ldots, n$), the colors of following columns are determined based on previous columns as follows:

(1) A new set of columns is created by:

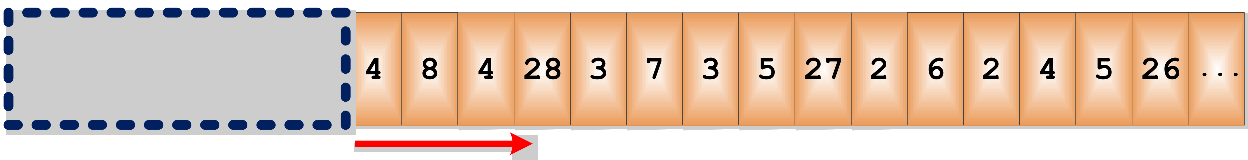
1a. taking the previous set of columns with the reduction of 1 in the values of all columns

1b. remove all columns with value 0

1c. add a column to the end with the value equal the number of the columns of the previous set

(2) Append the new set of columns to the existing sequence of columns.

For example, with $n = 4$, $m = 6$, and the initial set of columns is (30,2,5,9), the next set of columns is (29,1,4,8,4), then the set (28,3,7,3,5), etc. The sequence of columns in that case looks like:



The researchers have tested different initial sequences of columns to find the optimal column sequence for best treatment effect. The dedication and enthusiasm of the research team are so admirable. A large number of initial column sequences are tested. Finally they found the optimal sequence, which they are going to present at an international conference. Unfortunately, the absent-minded secretary of the group forgot the color of the last column when noting down this initial sequence. Fortunately, there is a note in the report that describes two characteristics of the sequence. First, the sum of all colors in the sequence does not exceed 255. Second, from the position ***p*** to the position ***q***, there are exactly ***k*** pair of colors c_i and c_j satisfying the inequality $c_j < c_i$ for $p \le i < j \le q$.

Given ***n***, c_i ($i = 1, 2,\ldots,n-1$), ***p***, ***q*** and ***k***, your task is to write a program to determine the minimum c_n satisfying the mentioned requirements.

Input

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

For each data set, there are 1 or 2 lines depending on the value of n. The first line contains 4 integers n, p, q and k separated by spaces ($1 \le n \le 255$, $1 \le p \le q \le 10^9$, $0 \le k \le (q - p)\times(q - p + 1)/2$). If $n > 1$, the data set contains a second line with n -1 integers $c_1, c_2, \ldots, c_{n-1}$ separated by spaces ($1 \le c_i \le 255$, $i = 1,2, \ldots n-1$).

Output

For each data set, write in one line the number c_n if found or -1 otherwise.

Sample Input	Sample Output
2 5 10 18 17 4 8 5 1 7 41 50 16 8 5 1 4 2 6	3 4

Problem E: Hanoi Tower

In the very modern game of Hanoi Tower, there is a stack of n disks having different sizes placed on top of each other. You are collaborating with a robot by making alternating moves to sort the disks in order (the smallest disk is on top and the largest disk is at the bottom). Each move is a swap of two consecutive disks. The robot is pretty dumb, so it just takes a pure random move (i.e., every consecutive pair of disk has an equal probability to be swapped). Knowing the robot's strategy, you try to act to minimize the expected total number of moves to sort the disks in the required order. In other words, the average total number of moves to sort the disks across all possible cases is minimized.

Your task is to write a program to determine the expected total number of moves to sort the disks.

Input

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

Each data set comprises of 2 lines. In the first line, there is a single number n ($n \le 8$), the number of disks. In the second line, there are n different integers d_i (i=1...n) separated by space describing the size of the disks from top to bottom ($1 \le d_i \le n$).

Output

For each data set, write in one line the expected number of moves to reach the goal, rounded to the precision of 5 decimal digits.

Sample Input	Sample Output
1 3 3 1 2	4.00000

Problem F: Protein secondary structure prediction

Proteins are one of the most important biological macromolecules in all organisms. Each protein is formed from 20 possible amio acids abbreviated by 20 different letters: A, R, N, D, C, E, Q, G, H, I, L, K, M, F, P, S, T, W, Y, V. In this problem, we consider two types of protein structure: primary structure and secondary structure. The primary structure **X** of a protein **O** is described by a sequence of 20 mentioned amio acids. The secondary structure **Y** of a protein **O** describes how amino acids in the primary structure of **O** are linked by hydrogen bonds. Each amino acid in the primary structure can belong to one of three possible secondary structure types: alpha helix (A), beta sheet (B), and Turn (U). **Y** is formed by the sequence of the secondary structure types of the corresponding amino acids in **O**'s primary structure. The following figure shows an example of the primary and secondary structure of a protein with 16 amino acids.

Position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Primary structure	A	R	N	D	F	Y	Y	Y	A	D	A	S	C	E	F	D
Secondary structure	A	A	A	A	U	U	U	B	B	B	B	B	A	U	B	B

Let **X** = ($x_1, x_2, \ldots, x_n$) be the primary structure of a protein **P** with n amino acids where x_i is one of twenty possible amio acids. Let $Y = (y_1, y_2, \ldots, y_n)$ be the secondary structure of that protein where y_i is one of three secondary structure types. A position i ($1 < i < n$) on **Y** is called *unnatural point* if $y_{i-1} \neq y_i$ and $y_i \neq y_{i+1}$. A secondary structure **Y** is called *natural* if it contains less than k *unnatural points*.

The secondary structure of a protein is not easily determined by biological experiments. They are usually predicted from its primary structure. The score that amino acid x belongs to secondary structure type y is **Q**(x, y) as shown in the following table.

Q(x,y)	Alpha helix (A)	Beta sheet (B)	Turn (U)
A	142	83	66
R	98	93	95
N	101	54	146
D	67	89	156
C	70	119	119
E	151	37	74
Q	111	110	98
G	57	75	156
H	100	87	95
I	108	160	47
L	121	130	59
K	114	74	101
M	145	105	60
F	113	138	60

P	57	55	152
S	77	75	143
T	83	119	96
W	108	137	96
Y	69	147	114
V	106	170	50

Given the primary structure **X** of protein **O** with ***n*** amino acids, the score ***p*** that **Y** is the secondary structure of protein **O** is calculated as following:

Given the primary structure **X** of a protein **O** with ***n*** amino acids ($0 < n < 10^6$), your task is to write a program to find the *natural* secondary structure **Y** of **O** with the highest score.

Input

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

Each data set consists of two lines. The first line contains an integer ***k*** ($0 < k < 1000$). The next line contains a string representing the primary structure **X** of protein **O**.

Output

For each data set, write in one line the score of the *natural* secondary structure **Y** of **O** with the highest score.

Sample Input	Sample Output

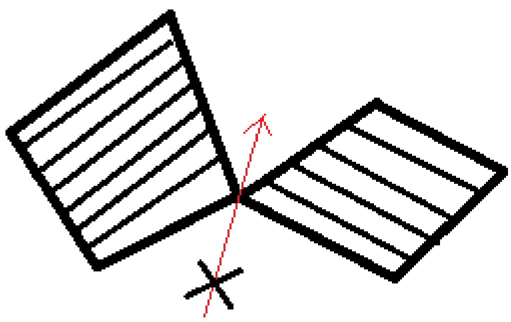
`1`
`1`
`AN`
	`288`

Problem G: Urbanization

While building a new city in an empty land of MCA, they have to design and build a certain number of buildings. For each building, they have several designs: electricity infrastructure design, water infrastructure design, etc. Each design occupies an area having a shape of a convex polygon. Because the designs of each building do not occupy exactly the same area, they have decided to construct the building in the area that is the intersecting area of all its designs. Assume that the areas for all buildings do not overlap but can share border.

Linh, a young and enthusiastic worker, decided to move to the new city of MCA. With no vehicle, every day he has to walk to work. Of course, they do not allow him to walk across buildings. Linh can walk along the border of the buildings but cannot cross any border or segment created by joining borders.

The following figure illustrates a case when Linh cannot walk through the joining borders of two buildings.



Your task is to write a program to determine the shortest way for Linh to get from his home to his office assuming that it always exists.

Input

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

Each data set consists of a number of lines. The first line contains five integer n, x_1, y_1, x_2, y_2 separated by space ($1 \le n \le 20$, $0 \le x_1, y_1, x_2, y_2 \le 10000$), where n is the number of the buildings, (x_1, y_1) is the coordinate of Linh's home, and (x_2, y_2) is the coordinate of Linh's office. The following lines describe the designs of n buildings. For each building, the first line contains m_i ($1 \le m_i \le 5$), which is the number of designs for the $\mathbf{i}^{th}$ building ($1 \le \mathbf{i} \le n$). Each of the next m_i lines contain a number t_{ij} ($1 \le t_{ij} \le 20$), which is the number of vertices of the polygon representing the $\mathbf{j}^{th}$ design ($1 \le \mathbf{j} \le m_i$) of the $\mathbf{i}^{th}$ building, followed by t_{ij} pairs of integers x,y representing the coordinates of the vertices of the polygon in order. All numbers are separated by a space.

Output

For each data set, write in one line the integer part of the length of the shortest way to get from Linh's home to his office.

Sample Input	Sample Output

1 2 0 0 0 4 2 4 1 1 1 2 2 2 2 1 4 1 1 1 2 2 2 2 1 1 4 2 2 2 3 3 3 3 2	4

Problem H: Digits

Let's denote ***S*** to be the string of infinite length formed by concatenating all positive integers together in increasing order:

S = 1234567891011121314151617181920….

Given a digit ***d***, a count ***k***, your task is to write a program to find the index (starting from 1) where the string formed by concatenating ***k*** digit ***d*** first appears in ***S***.

Input

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

Each data set consists of only one line, which contains two integers ***d*** ($0 \le d \le 9$) and ***k*** ($1 \le k \le 10^4$) separated by space.

Output

For each data set, write in one line the index where the string of ***k*** digit ***d*** first appears in ***S***.

Sample Input	Sample Output
`1` `1 3`	`12`

Problem I: Magic Sequence

Center of Genius is a special research center attracting famous scientists from all over the world. Luckily you have been offered to work at the center as an intern for 6 months. There is a secure area that you would like to visit where they showcase many state-of-the-art technologies, each in a separate room. To enter a room in this secure area, you need to solve one specific puzzle.

The security system randomly generates a **challenge sequence** containing $N - 1$ symbols $s_1 \ldots s_{n-1}$ where s_i is either '<' or '>'. You are required to find a **key sequence** of N distinct integers in the range from 1 to N $a_1, \ldots, a_n$ such that the inequality $a_i\ s_i\ a_{i+1}$ is satisfied for all $i = 1,2,\ldots,N\text{-}1$. Please remember that each integer from 1 to N must be used only once in the key sequence.

For example, with the challenge sequence <<>< and the key sequence 1 2 4 3 5, all inequalities are satisfied:

$$1 < 2 < 4 > 3 < 5$$

Your task is to write a program to find the **key sequence** mentioned above. If there is more than one valid key sequence, you should select the smallest one in lexicography order.

Input

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

Each data set consists of two lines where the first line contains number N ($N \le 10^5$). The second line contains the challenge sequence of N-1 characters.

Output

For each data set, write in one line the corresponding space-separated key sequence.

Sample input	Sample output
1 5 <<><	1 2 4 3 5

Problem J: Parentheses

A non-empty string containing only parentheses is considered *balanced* if it has the same number of "(" and ")", and for any prefix of , the number of "(" is greater or equal to the number of ")". For example, the following strings are balanced:

```
()(())
((()))
```

while the following strings are not balanced:

```
)(
(((()))
)()()(
```

A string is said to be a *substring* of a string if is matched with a consecutive sequence of characters in .

Given a string , your task is to write a program to count the number of *different* balanced substrings of .

For example, string ***T*** = (()())() has 4 different balanced substrings: (), ()(), (()()), (()())(). String ***T*** = (()()() ()() has 5 different balanced substrings: (), ()(), ()()(), ()()()(), ()()()()().

Input

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

Each data set consists of one line containing a non-empty string of parentheses (0 < ≤).

Output

For each dataset, write in one line the corresponding number of different balanced subtrings.

Sample Input	Sample Output
3 (()())() (()()()()() ()()()(()())(()())	4 5 11