

# National Girls' Programming Contest 2018



22<sup>nd</sup> October 2018

You get 19 Pages

10 Problems



Hosted by

**Daffodil International University**

**A**

# Welcome to NGPC

**Input:** Standard Input  
**Output:** Standard Output

This year Daffodil International University is hosting National Girls Programming Contest - 2018. Two pro-active person - Anup Majumder and Saiful Islam, have worked hard to make this event a great success.

A very important thing of this event was to print banners to welcome all of you guys. To make this work easy, they nominated two of their students as volunteer to complete this task. After the students were assigned with the task, they were talking about the place from where they should print the banner. One student was saying, "We can print this from nilkhet at a very low price". The other student denied that and said, "No, we can print the banner from gulistan cheaper than nilkhet" and they started arguing on that topic.

At a stage, the students didn't come to a single opinion and they went to complete their task at their own. Both of them wanted to prove them right. So, both of them printed a banner and took those to their teachers. Their teachers were very shocked seeing this! Because a single banner printed with a bit more cost is okay but now they have to pay for both of the banners. Anup Majumder and Saiful Islam came to know that one of the banners costs two thousands taka and the other one costs three thousands taka. In total now they had to pay both amount for this purpose. Now as a setter of this problem, I am assigning you with a task, can you just print the total amount of money the hosts spent for printing the banner?

## Input

This problem has no input.

## Output

Just print the summation of costs. Don't forget to print a new line.

## Sample Output

200

**N.B:** This output is not the actual one. Please find the output from our story.

# B

## GRE Exam

**Input:** Standard Input  
**Output:** Standard Output

You must have heard about GRE exams. The full form of GRE is Graduate Record Examinations. This is required for many universities around the globe for graduate level admission. Many of you may take GRE in the future, so I thought I will give you a preview of the exam.

The computer-based GRE General Test consists of six sections. The first section is always the analytical writing assessment (AWA) section involving separately timed issue and argument tasks. The next five sections consist of two verbal reasoning sections, two quantitative reasoning sections, and an experimental section.

These five sections may occur in any order. The experimental section can be either verbal or quantitative. The experimental section does not count towards the final score but is not distinguished from the scored sections. For our problem we will ignore the AWA section.

Each of the five sections has 20 questions. You get 1 marks for each correct answer and 0 otherwise. For some reason, the scoring starts from 130 for each part (verbal and quantitative). So, if you get all the question wrong, then the minimum score you can get is 260 (130 in each part). If you answer all the question right, then you can get a maximum of 340 (170 in each part).

You will be given the information about five sections (ignoring AWA) of a participant. For each section, we will mention whether it is verbal or quantitative. We will also mention if it's experimental or not. Then for each section, we will mention how many questions was answered correctly. Based on these information, you have to calculate the final score of the participant.

### Input

Input contains of description for **5** sections in **5** lines. Each section will be described using **3** integers **S**, **T** and **C**. **S** will be **0** or **1**, where **0** means it's a verbal section and **1** means, it's a quantitative section. **T** will be **0** or **1**, where **0** means it's regular and **1** means it's experimental. **C** is the number of correctly answered questions on that section, which will be between **0** and **20**. Exactly one section will be experimental.

### Output

Output three lines. First line will contain the total score in verbal sections. Second line will contain the total score in quantitative sections. Third line will contain overall total score. See sample for clarifications. Don't forget to put a newline, after each line.

### Sample Input 1

```
0 0 10
0 1 15
1 0 18
1 0 19
0 0 12
```

### Output for Sample Input 1

```
Verbal: 152
Quantitative: 167
Total: 319
```

### Sample Input 2

```
1 1 20
0 0 10
1 0 10
0 0 20
1 0 0
```

### Output for Sample Input 2

```
Verbal: 160
Quantitative: 140
Total: 300
```

# C

## Sam the Newspaper Guy

Input: Standard Input  
Output: Standard Output

Sam distributes newspapers in his village. The village has  $n$  houses numbered from  $1$  to  $n$  and they are located by cartesian point  $(x, y)$ . Every morning Sam starts his journey from his home (which is the house number  $1$ ) and distributes newspapers serially according to the house numbers and then returns back home. The cost of travel between two houses is the square of their euclidean distance. The total cost of a journey is the summation of all the costs.

In his journey, he never visits the same house twice except his own. In other words, everyday he completes a cycle. By doing the same journey, again and again, he gets bored, so on his path, he calculates the total cost of his journey. One unique thing about the village is, everyday position of exactly one house permanently changes.

You are given  $n$  coordinates representing positions of houses serially from  $1$  to  $n$ . Then you are given  $m$  queries, where each query represents the house number and its new position. For each query calculate the total cost of the journey.

### Input

First line contains an integer  $T$  (number of test cases) ( $1 \leq T \leq 5$ ). For each test case, the corresponding first line contains two integers representing  $n$  (number of houses) ( $2 \leq n \leq 10^5$ ) and  $m$  (number of queries) ( $1 \leq m \leq 10^5$ ). Followed by  $n$  lines, where  $i$ 'th line contains two integers  $x, y$  ( $-10^5 \leq x, y \leq 10^5$ ) representing coordinates of  $i$ 'th house. Followed by  $m$  lines, where  $i$ 'th line contains three integers representing the house number that moved and coordinates of new position. **Input file is huge. Please use faster I/O.**

### Output

In first line of every test print the test case number. Then for each query, print the cost in a new line. See the sample input output to see the format.

### Sample Input

```
1
3 4
0 0
1 5
-1 6
2 1 0
3 1 5
1 -1 -1
3 -3 2
```

### Output for Sample Input

```
Case 1:
78
52
70
38
```

## Explanation

In the first query, the locations of houses are:

House-1 located at (0, 0)

House-2 located at (1, 0)

House-3 located at (-1, 6)

The cost to travel from House-1 to House-2 is 1

The cost to travel from House-2 to House-3 is 40

The cost to travel from House-3 to House-1 is 37

Total cost is  $(1 + 40 + 37) = 78$

In the second query, the locations of houses are:

House-1 located at (0, 0)

House-2 located at (1, 0)

House-3 located at (1, 5)

The cost to travel from House-1 to House-2 is 1

The cost to travel from House-2 to House-3 is 25

The cost to travel from House-3 to House-1 is 26

Total cost is  $(1 + 25 + 26) = 52$

In the third query, the locations of houses are:

House-1 located at (-1, -1)

House-2 located at (1, 0)

House-3 located at (1, 5)

The cost to travel from House-1 to House-2 is 5

The cost to travel from House-2 to House-3 is 25

The cost to travel from House-3 to House-1 is 40

Total cost is  $(5 + 25 + 40) = 70$

In the fourth query, the locations of houses are:

House-1 located at (-1, -1)

House-2 located at (1, 0)

House-3 located at (-3, 2)

The cost to travel from House-1 to House-2 is 5

The cost to travel from House-2 to House-3 is 20

The cost to travel from House-3 to House-1 is 13

Total cost is  $(5 + 20 + 13) = 38$

# D

## Change of Apartment Again

Input: Standard Input  
Output: Standard Output

This time Mesbah has got a new job and he will shift to a new apartment which is close to his office.

His current apartment is a bit weird. It can be modeled as a  $1 \times n$  grid. The leftmost cell is numbered as 1 and the rightmost cell is numbered as  $n$ . Look at the image below for clarification.



For each  $i$  ( $0 < i < n$ ), there is a door between cell  $i$  and cell  $i + 1$ . Mesbah can move furniture from one cell to another through these doors. There is an **exit door** located at cell 1 which is used to move furniture out of the house. He is allowed to create exactly one **new exit door** at some cell  $x$ . His goal is to move all the furniture out of the house through the exit doors.

In one second he can move only one furniture to the cell left to it or to the cell right to it. Time needed to move the furniture out of the house through any exit door is also one second provided that the furniture is currently in cell 1 or in cell  $x$ .

Note that, a cell can contain multiple pieces of furniture in this ongoing process. When he is not moving a furniture, he can step to any cell without consuming any significant amount of time. And initially, there is at most one furniture in each cell.

Mesbah wants to finish the movement of furniture **in the minimum time possible**. As many of you are experts in this kind of task, he is asking for your help. You have to select an optimal position  $x$  for the new **exit door** such that the total time needed to move all the furniture out of the apartment is minimized.

Note that, there can be many positions for the new exit door which minimize the needed time. So you are asked to output the **minimum needed time** instead.

### Input

The first line of input contains an integer  $T$  ( $1 \leq T \leq 100$ ), the number of test cases. Every test case starts with a line containing an integer  $n$  ( $2 \leq n \leq 5 * 10^5$ ) denoting the dimension of the apartment. The following line contains a string  $S$  of size  $n$ . If there is a furniture at cell  $i$  then  $S_i = 1$  else  $S_i = 0$ .  
**Input file is huge. Please use faster I/O.**

## Output

For each test case, output a single line containing the test case number and the minimum possible time needed to move all the furniture out of the apartment. Follow the sample output for details.

### Sample Input

### Output for Sample Input

2 5 10101 7 1001101	Case #1: 5 Case #2: 7
---------------------------------	--------------------------

## Explanation

For **Case 2** of the sample input, one of the optimal positions for the new door is **cell 5**.

If we build the new door at cell 5,

Furniture at cell 1 will be moved out through cell 1 ( Time needed =  $0 + 1 = 1$  )

Furniture at cell 4 will be moved out through cell 5 ( Time needed =  $1 + 1 = 2$  )

Furniture at cell 5 will be moved out through cell 5 ( Time needed =  $0 + 1 = 1$  )

Furniture at cell 7 will be moved out through cell 5 ( Time needed =  $2 + 1 = 3$  )

Total needed time =  $1 + 2 + 1 + 3 = 7$

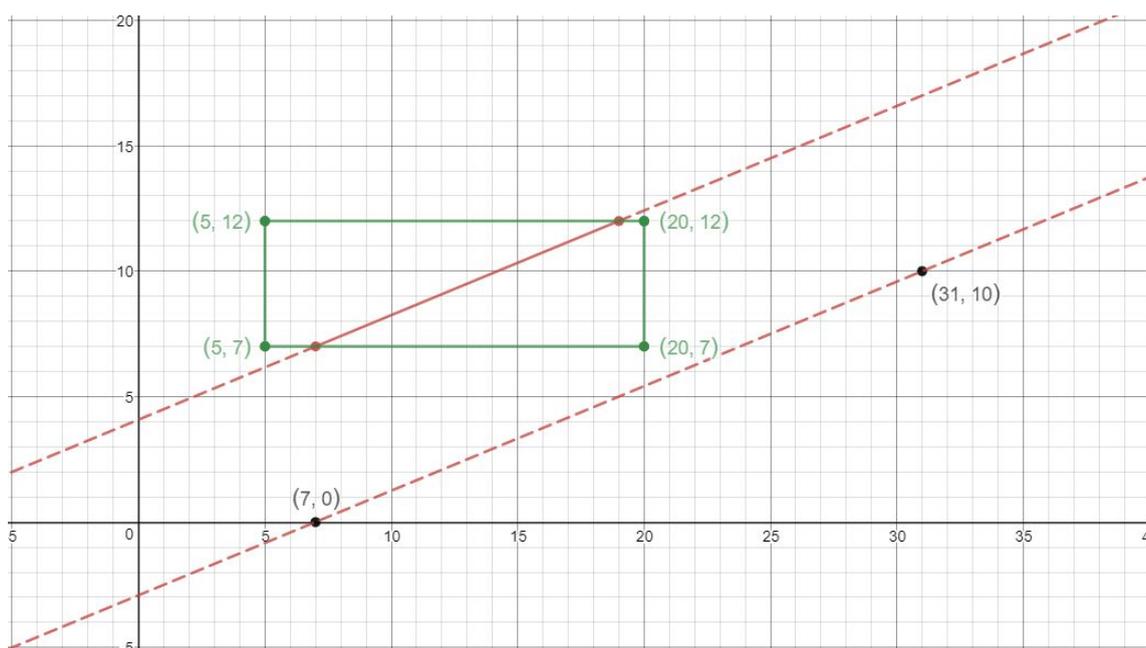
# E

## An Easy Geometry Problem Maybe!

Input: Standard Input  
Output: Standard Output

Recently I have come up with a geometry problem. As usual, I haven't been able to solve the problem yet. I hope you can help me solve it.

In this problem, you will be given a rectangle and a line. The sides of the rectangle will be axis parallel. Some portion of the line might be inside the rectangle. Let  $L$  denote the length of that portion. The line can be moved parallel to itself. Your target will be to maximize the value of  $L$ .



Description of the Second Test  
Case of The Sample Input

### Input

First line of the input contains an integer  $T$  ( $1 \leq T \leq 10^4$ ) which denotes the number of test cases. Each of the following  $T$  lines contain 8 integers  $X1, Y1, X2, Y2, X3, Y3, X4$  and  $Y4$  ( $-10^4 \leq X1, Y1, X2, Y2, X3, Y3, X4, Y4 \leq 10^4$ ). Point  $(X1, Y1)$  denotes the lower left corner and point  $(X2, Y2)$  denotes the upper right corner of the rectangle. The given line goes through point  $(X3, Y3)$  and point  $(X4, Y4)$ .

The side lengths of the rectangle will always be positive ( $X1 < X2$  and  $Y1 < Y2$ ). The two points defining the line will never be same points.

**Input file is huge. Please use faster I/O.**

## Output

For each test case, output a single line containing the test case number and the maximum possible value of **L** rounded to 4 decimal places. Follow the sample output for details.

### Sample Input

### Output for Sample Input

2 0 0 8 4 5 0 11 8 5 7 20 12 7 0 31 10	Case #1: 5.0000 Case #2: 13.0000
--	-------------------------------------

# F

## NGDC Teams

**Input:** Standard Input  
**Output:** Standard Output

Finally, after so many years, the National Girls Dance Contest is going to be held on October 22, 2018. All the teams from each university are preparing for the final battle to show their impressive skills. Being the dance faculty of UpsideDown University, Miss Youkee is planning to form some strong teams for the competition to ensure the top prizes for her University. But she doesn't yet know all of her students very well. So before forming the teams for NGDC, she called all the **N** students in her university to attend a practice session on a fixed date.

But the girls have their academic studies including their midterms, quizzes, assignments and so many stuffs! So Miss Youkee has announced that each student may enter the practice session any time they find convenient and they might also leave whenever they want. But each student must attend the practice session exactly once on the given fixed date. So whenever a student leaves the session, she will never come back again.

Each of the students has got a **skill level** that signifies how good they performed in their previous competitions. Additionally, whenever a student enters the session, she will be provided with an **ID** which is equal to the number of students entered in the practice session so far including her. So the ID of the first student entering the practice session is 1, the ID of the second student is 2 and so on.

Whenever any student enters or leaves the practice session, Miss Youkee asks all the current students in the session to stand in a line. Then she forms some teams following a mysterious algorithm and asks the students to continue their practice together with their team until another student enters or leaves the session again. The mysterious algorithm of Miss Youkee follows the below steps.

Step 1: Sort all the students in the line in ascending order of their ID.

Step 2: If the line is empty then terminate the algorithm.

Step 3: Choose the longest prefix of the students whose skill level follows an increasing order.

Step 4: Form a team including all the students from that longest prefix.

Step 5: Remove all the students in that prefix from the line and continue to step 2.

For example, let's say there are currently 8 students in the session and after sorting them in ascending order of their ID their skill level looks like this: **[1, 5, 7, 9, 2, 4, 8, 3, 6]**. Following the mysterious algorithm of Miss Youkee, the below teams will be formed by the end of the algorithm.

Team 1: **[1, 5, 7, 9]** will be selected to form the first team. Because this is the longest prefix which follows a strictly increasing sequence. After forming this team, the prefix will be removed from the line. So the remaining line will now look like this: **[2, 4, 8, 3, 6]**.

Team 2: [2, 4, 8] will be selected to form the second team. Then this prefix will be removed and the line will look like this: [3, 6].

Team 3: [3, 6] will be selected to form the third team. Finally, after removing this prefix the line will become empty and Miss Youkee will terminate the algorithm.

So in this example, Miss Youkee will form exactly 3 teams. And those three teams will start practicing together until another student enters or leaves the session.

**Note:** The prefix of a line is a sequence of consecutive elements that starts from the beginning of that line.

As you are the technical manager of the university, Miss Youkee needs your help to keep track of the number of teams she formed after each of the incidents when a student entered or left the session. Because she'll need this information to form the final teams for the NGDC. So you'll be provided with the info of all those incidents and each time when a student entered or left the session, you need to find how many teams Miss Youkee formed considering the existing students in her session during that time.

## Input

The first line contains an integer  $T$  ( $1 \leq T \leq 10$ ) denoting the number of test cases. The first line of each test cases contains an integer  $N$  ( $1 \leq N \leq 50000$ ), the number of total girls in UpsideDown University. Each of the next  $2N$  lines contains two integers  $S$  ( $1 \leq S \leq 2$ ) and  $X$  ( $1 \leq 100000 \leq X$ ). When  $S$  is 1, it means a student with skill level  $X$  enters the practice session. Similarly, when  $S$  is 2, it means a student with skill level  $X$  leaves the practice session. The ID of the  $i$ 'th student entering the session is  $i$ . Besides, the skill level of the students are distinct and whenever a student leaves the session, she'll never enter the session again.

**Input file is huge. Please use faster I/O.**

## Output

After each of the incidents, print a single integer which denotes the number of teams that can be formed considering the existing students in the practice session following the mysterious algorithm of Miss Youkee.

## Sample Input

## Output for Sample Input

1	1
6	1
1 3	1
1 5	2
1 8	2
1 6	2
1 9	3
2 3	2
1 7	2
2 8	1
2 5	1
2 7	0
2 9	
2 6	

## Explanation

Let's consider the scenario after the first 7 incidents happened. The students with skill level 3, 5, 8, 6, 9 and 7 entered the practice session so far. But the student with skill level 3 left the session in the previous incident. So if we sort the remaining students in ascending order of their ID then their skill level will look like this: [5, 8, 6, 9, 7]. Now, exactly three teams can be formed considering the current students. [5, 8] will form the first team, [6, 9] will form the second team and [7] will form the third team. Hence, our 7<sup>th</sup> output is 3.

# G

## Full Binary Tree

Input: Standard Input  
Output: Standard Output

A full binary tree is a tree in which every node other than the leaves has **exactly two** children. Leaf node has zero children. Let **ID** of the root node is **1**. If **P** is the ID of node **U** and it is also not a leaf node, then because of the property of binary tree,  $(P * 2)$  is the ID of it's left child and  $(P * 2 + 1)$  is the ID of it's right child.

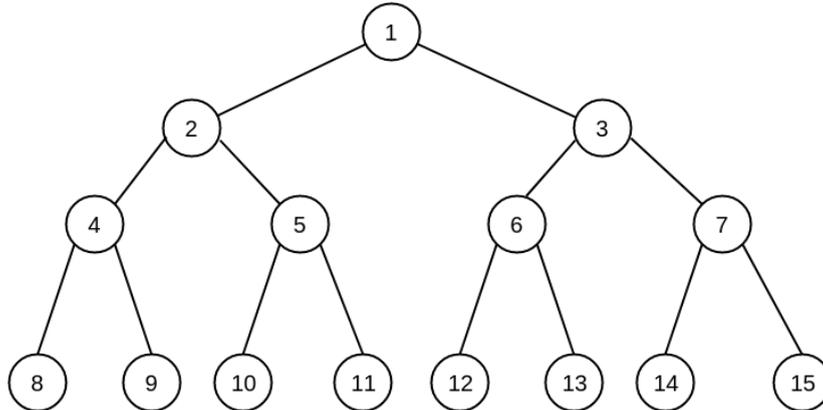


Figure: A full binary tree with 15 nodes

You are given a full binary tree and each node contains an integer value.  $\text{Cost}(U, V)$  is the sum of values of all nodes in the path from **U** to **V**. Your task is to find the such pair of nodes **(U, V)** such that  $\text{Cost}(U, V)$  is **maximum** where **U** and **V** should be **leaf nodes** and  $U \neq V$ . You don't need to find any specific pair of nodes but you have to find the maximum  $\text{Cost}(U, V)$ .

### Input

First line contains an integer **T** ( $1 \leq T \leq 20$ ), denoting the number of test cases. For each test case, first line contains an integer **N** ( $1 \leq N \leq 131071$ ) denoting the number of nodes. It is guaranteed that  $(N+1)$  will always be a power of **2**. Second line contains array **A** of **N** space separated integers denoting the values assigned to each nodes from **1** to **N** sequentially ( $|A[i]| \leq 10^5$ ). **Input file is huge. Please use faster I/O.**

### Output

For each test case, print the maximum **Cost** possible.

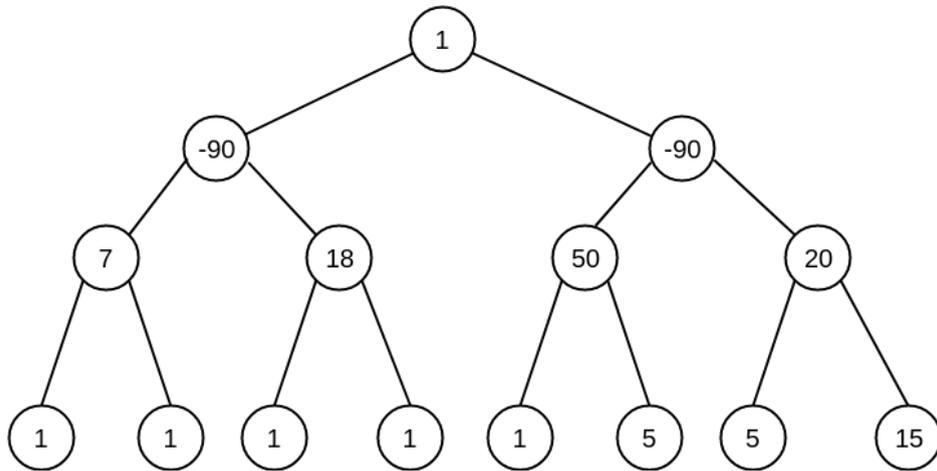
### Sample Input

### Output for Sample Input

```
1
15
1 -90 -90 7 18 50 20 1 1 1 1 1 5 5 -1
```

```
56
```

## Explanation



In the given sample test case -  $\text{Cost}(12, 13) = 1 + 50 + 5 = \mathbf{56}$ . This is the maximum Cost among all possible pairing of leaf nodes.

# H

## Shahin and His Twin Children

Input: Standard Input  
Output: Standard Output

All of you probably heard about the mystery of “Shahin and His apartment”. Today I’m going to share another mystery of his new life. This is all about his twin children Shahin Jr. and Rahin. The children like chocolate very much. To make them happy Shahin give exactly one chocolate each of them every day.

Riddho a friend of Shahin is a store-keeper. The name of the store is The Chocology. Every day at morning Riddho executes an operation. An operation defined with two integer  $X Y$ .

- When  $X = 1$  it means he brings infinity amount of chocolate of type  $Y$  to his store (because that type of chocolate is trending on other chocolate store and he does not have it). The type of chocolate is an integer number.
- When  $X = 2$  it means he removes all chocolate of type  $Y$  from his store. (because that type of chocolate is too outdated and people don’t like that anymore).

Shahin decided to buy two chocolates of different types for his children from the Chocology store at evening. Shahin does it every day as he wants to make his children happy. As Riddho and Shahin are friends Riddho offers Shahin a great deal. The deal is that, if Shahin buys two chocolate and type of this chocolate are  $t1$  and  $t2$  ( $t1$  and  $t2$  have to be available on chocology store on that day) then Shahin has to pay the amount of  $M$  money ( $M = t1 \wedge t2$ , here  $\wedge$  denotes bitwise **XOR** operator). But Shahin wants to do this optimally means he wants to buy two different type of chocolate paying as less money as possible. As we know you are a programmer we want you to help Shahin.

### Input

The first line of the input contains an integer  $T$  ( $1 \leq T \leq 20$ ). Denoting the number of a test case. Each test case starts with one integer  $N$  ( $1 \leq N \leq 10^5$ ).  $N$  denotes the number of the days we want to help Shahin. Next  $N$  lines describe the operation that are performed by Riddho. Each of the next  $N$  lines contain two integer  $X Y$  ( $1 \leq X \leq 2, 0 \leq Y \leq 10^{18}$ ). which define the operation of Riddho as described above. The first day he performs first operation second day perform the second operation and so on.

**Remember that Riddho performs his operation at morning and Shahin buys chocolate in the evening. You can assume that initially The Chocology store is empty. Input file is huge. Please use faster I/O.**

## Output

For each day print, the minimum amount of money Shahin has to pay in separate line. If he cannot buy two different type of chocolate then print -1. See sample output below for exact formatting.

### Sample Input

### Output for Sample Input

3	Case #1:
3	-1
1 1	3
1 2	-1
2 2	Case #2:
3	-1
1 1	3
1 2	1
1 3	Case #3:
3	-1
1 1	-1
2 1	-1
1 1	

## Explanation

### First Case:

<i>Day</i>	<i>Available Chocolates</i>	<i>Best Pair</i>	<i>Minimum Price</i>
1	1	No pair	-1
2	1,2	1,2	$1^2 = 3$
3	1	No pair	-1

### Second Case:

<i>Day</i>	<i>Available Chocolates</i>	<i>Best Pair</i>	<i>Minimum Price</i>
1	1	No pair	-1
2	1,2	1,2	$1^2 = 3$
3	1,2,3	2,3	$2^3 = 1$

# I

## Ugly Function

**Input:** Standard Input  
**Output:** Standard Output

You are given a string **S** of lowercase alphabet. You have to generate a new string **NS**. In order to generate the new string **NS**, we have a piece of code which contains two functions:

```
FindMaxValue(int A, int B){
    /*
    This function returns the maximum value for the following equation for
    any values of x:
    y = A*A + B*x - (27-B)*x*x
    */
}

Convert(string S){
    string NS;

    for (int i=0;i<len(S);i++){
        int a = S[i] - 96;
        int y = FindMaxValue(i, a);
        int b = ((y%101) % 26) + 1;

        NS += (char)(b + 96);
    }
}
```

Though we have the piece of code, but unfortunately we don't know how to implement the FindMaxValue function. So, you have to help us implementing that function and give us the generated string **NS**.

**N.B:** The value of **y** in Convert function can be a fraction of the form **p/q**. But **(p/q) MOD 101** will be an integer.

### Input

The first line of the input contains an integer **T** ( $1 \leq T \leq 100$ ), denoting the number of a test case. Each case contains a string **S** ( $1 \leq |S| \leq 10^5$ ). **Input file is huge. Please use faster I/O.**

### Output

For each case, print the string **NS**.

### Sample Input

```
2
abcd
zyxw
```

### Output for Sample Input

```
jakd
rtfq
```

# J

## Prime Digit

Input: Standard Input  
Output: Standard Output

You will be given a number  $N$ . You have to find the number of **PRIMEDIGIT** numbers between 1 to  $N$ . A number  $X$  is called **PRIMEDIGIT** number if the number of occurrences of each digit in  $X$  is a prime number.

For example, the number  $X = 44666$  is a **PRIMEDIGIT** number. Because the digit 4 occurs 2 times and the digit 6 occurs 3 times in  $X$ . As both frequencies 2 and 3 are prime numbers so  $X$  is a **PRIMEDIGIT** number. But the number  $Y = 377$  is not a **PRIMEDIGIT** number. Because the frequency of the digit 3 in  $Y$  is 1 which is not a prime number.

### Input

The first line contains  $T$  ( $1 \leq T \leq 25$ ) the number of test cases. Each of the next  $T$  lines contains a single integer  $N$  ( $1 \leq N \leq 10^{18}$ ).

### Output

For each test case print a single integer, the number of **PRIMEDIGIT** numbers between 1 to  $n$ .

### Sample Input

```
3
50
100
500
```

### Output for Sample Input

```
4
9
13
```

### Explanation

The **PRIMEDIGIT** numbers between 1 to 50 are 11, 22, 33, 44.