

# 10<sup>th</sup> IIUC Inter-University Programming Contest

I I U P C 2 0 1 3



## Problemset

*Allah grant me the serenity  
To accept the problems that I cannot solve  
The persistence to solve the problems that I can  
And the wisdom to know the difference*

**Organized by: Computer Club  
International Islamic University Chittagong**

## Problem A: See Emily Play

You are taking care of Emily this afternoon. To keep her entertained, you've designed a game for her involving some math. The rule for the game is such --

Emily is initially given a big number  $N$ . In each step she is supposed to:

1. If  $N$  is  $0$ , stop and call it a day
2. If  $N$  is divisible by  $2$ , divide  $N$  by  $2$ .
3. Otherwise, subtract  $1$  from  $N$ .

For example, starting with  $14$ , she gets  $7, 6, 3, 2, 1, 0$

Now, you see, Emily is just a kid and hasn't grown strong notions of rules yet. So, as you have given the rules to her, Emily tries, but misunderstands. In her mind, she rewrites the rules :

1. If  $N$  is  $0$ , stop and call it a day
2. If  $N$  is divisible by  $2$ , divide  $N$  by  $2$ .
3. Randomly and unbiasedly choose to do any of these two :
  - subtract  $1$  from  $N$
  - add  $1$  with  $N$

For each division Emily takes  $d$  seconds, For each subtraction she takes  $s$  seconds, for each addition she takes  $a$  seconds. Now that you know how she is bending the rules, you start to wonder how many seconds you have to see Emily play.

### Input

First line of input will contain the number of test cases,  $T \leq 100000$ . Then  $T$  test cases follow. For each case there will be a single line containing four integers separated by space :

$N \ d \ s \ a$

where ,

$1 \leq N \leq 1000000$

$1 \leq d, s, a \leq 10$

### Output

For each case output a single line containing one real number, the expected number of seconds :  $T$ . Show exactly  $3$  digits after decimal point, properly rounded.

Sample Input	Output for Sample Input
6	3.000
1 1 1 1	4.000
2 1 1 1	5.500
3 1 1 1	5.000
4 1 1 1	6.750
5 1 1 1	29.367
1000000 1 1 1	

**Problem Setter : Pratyai Mazumder**

**Alternate Solution : Md. Mahbulul Hasan, Prasanjit Barua**

## Problem B: Banglawash

*Banglawash!* It's a popular term used by the fans of Bangladesh cricket team when their team achieved a rare but much expected clean sweep over their opponent. In this November, 2013, the touring New Zealand cricket team is once again *Banglawashed* in the 3-match ODI series.

In cricket, the term *whitewash* is used when one team wins all the matches played in a particular series; obviously abandoned match are not counted. Apart from this year's achievement, Bangladesh defeated New Zealand 4–0 to win a 5-match ODI series (one game was abandoned) in October 2010. New Zealand was touring Bangladesh on both the occasions. These two series were labeled as *Banglawash* as the language of Bangladesh is Bangla and 'Bangla' often used among locals for the things made in Bangladesh.

Some of the recent examples of *whitewash* include the following.

England's 4-0 defeat of India to win the Pataudi Trophy during India's tour of England in 2011.

Australia's 4-0 defeat of India to win the 2011-12 Border-Gavaskar Trophy.

India's 4-0 defeat of Australia to win the 2012-13 Border-Gavaskar Trophy.

West Indies' consecutive 5–0 defeats of England in 1984 and 1985-86. These two results are also commonly labeled as *blackwashes* because of the dark skin of the West Indies players.

For the cricket enthusiastic fan of the Tigers, here is a list of eight *Banglawash*.

Opponent	Year	Result
Kenya	2006	4 – 0
Kenya	2006	3 – 0
Zimbabwe	2006	5 – 0
Scotland	2006	2 – 0
Ireland	2008	3 – 0
West Indies	2009	3 – 0
New Zealand	2010	4 – 0
New Zealand	2013	3 – 0

Now, Bangladesh is playing against the Rest of the World team, which is formed by taking players from World Wide Web (WWW) of cricketers. Hence, the team name is WWW.

### Input

First line of the input contains one positive integer **T** the number of test cases. The first line of each test case contains one positive integer **N**. **N** denotes how many match have been played in a single series. In the next line, there will be **N** uppercase letters. These letters will be either **B** or **W** or **T** or **A**. Here, **B** means the match was won by Bangladesh, **W** means the match was won by WWW, **T** means the match was a tie and **A** means the match was abandoned.

### Constraints

- **T < 101**
- **N < 11**

## Output

For each test case output the final outcome of the series. The outcome can be any one of the following.

**BANGLAWASH** – If Bangladesh won all the matches played (excluding the abandoned matches) in the series

**WHITEWASH** – If WWW won all the matches played (excluding the abandoned matches) in the series

**ABANDONED** – If all the matches of the series is abandoned

**DRAW** – If Bangladesh and WWW won equal number of matches in the series when the series is not abandoned

**BANGLADESH** – If Bangladesh won more matches then WWW in the series but failed to achieve a Banglawash

**WWW** – If WWW won more matches then Bangladesh in the series but failed to achieve a Whitewash

For BANGLADESH or WWW you have to show the number of match won by each country. And for DRAW you have to show how many wins and how many ties by each country.

For exact format see the sample output.

Sample Input	Output for Sample Input
6	Case 1: BANGLAWASH
3	Case 2: WHITEWASH
BBB	Case 3: BANGLADESH 2 - 1
3	Case 4: WWW 2 - 1
WWW	Case 5: DRAW 1 1
3	Case 6: ABANDONED
BWB	
4	
BWWT	
3	
BTW	
2	
AA	

**Problem Setter : Mohammed Shamsul Alam**

**Alternate Solution : Tanveer Ahsan**

## Problem C: The Twin Head Dragon

The Scourge are marching South-West with the biggest army ever seen, and they're marching fast. All the Sentinel towers are in ruins. There's chaos all over their base. Whatever they have to do, they have to do it by tonight or they will be terminated from the face of the earth tomorrow.

A secret meeting is being conducted by Zeus, the lord of Olympus and the father of gods. "The end is near." dreads Sven. "Careful child. Sentinel will not be doomed before my eyes." says Zeus. "Send Riki to explore the enemy camps. Then we can come up with a plan."

Riki comes with the information that there are  $N$  enemy camps and  $N-1$  bi-directional roads, each road connecting two camps. The lengths of the roads are different. Riki also found that there exists a path between any pair of camps.

"That's enough information!" says Zeus with excitement, "We have to burn down all the roads, so the enemies will be isolated from each other. Then we will strike. Summon Jakiro, he'll know what to do."

Jakiro, the Twin Head Dragon is summoned. He will use his ultimate spell, Macropyre to burn all the roads down. To do this he will follow these steps:

- 1) Select 2 different camps such that the shortest path between them doesn't include any burned road.
- 2) Prepare the spell with required mana. The mana cost for this spell is equal to the average of the lengths of roads in the path.
- 3) Burn all roads in the selected path.

He will keep burning this way until all the roads are burning. It is important that he uses minimum total mana for this task, as he needs mana for the battle afterwards. Now write a program to calculate the least mana required by Jakiro to burn all the roads down. Remember, you don't need to minimize the number times the spell is used.

### Input

The input will contain multiple test cases and number of test cases  $\leq 50$ . Each case starts with an integer  $N$  ( $2 \leq N \leq 14$ ) denoting the number of enemy camps. The camps are numbered from  $0$  to  $N-1$ . Each of the next  $N-1$  lines contain three integers  $A B C$  ( $0 \leq A, B < N, A \neq B, 1 \leq C \leq 10000$ ) denoting that camp  $A$  and  $B$  are connected by a road whose length is  $C$  units. You may assume that all pairs of  $A$  &  $B$  are unique. The input terminates with a value  $0$  for  $N$ .

### Output

For each case, print on a line the least total mana required by Jakiro rounded upto exactly 4 decimal points.

Sample Input	Output for Sample Input
4 0 2 1 1 2 2 2 3 3 6 0 1 10000 0 2 10000 0 3 1 0 4 1 0 5 10000 0	3.5000 15001.5000

### Output Explanation

In **sample test 1**, if we first select camps 1 & 3 and use spell on the path 1 – 2 – 3, the required mana would be  $(2+3)/2 = 2.5$ . Then we select camps 0 & 2 and use spell on the path 0 – 2 which would require  $1/1 = 1$  mana. So the total mana required to burn all the roads is  $2.5 + 1 = 3.5$ , which is the minimum value possible.

In **sample test 2**, we can use spell on paths 1 – 0 – 2, 0 – 3 and 4 – 0 – 5. It would cost mana  $(10000+10000)/2 = 10000$ ,  $1/1 = 1$  and  $(10000+1)/2 = 5000.5$  respectively. In total it would cost  $10000 + 1 + 5000.5 = 15001.5$  mana, which is the optimal value.

**Problem Setter : Sakib Hasan Sauro**

**Alternate Solution : Ashiqul Mostofa**

## Problem D: Dilation

Sabbir is a student of fourth year in CSE department. His course teacher recently gave him an assignment that is discussed below. But he is a little bit weak in programming. So, he needs your help to solve the problem.

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. One of the most basic morphological operations is *dilation*. *Dilation* adds pixels to the boundaries of objects in an image. The number of pixels added to the objects in an image depends on the size and shape of the *structuring element* used to process the image. A *structuring element* is a shape mask used in the basic morphological operations. They can be of any shape and size that is digitally representable, and each has an origin. The matrix dimensions specify the *size* of the structuring element and the pattern of ones and zeros specifies the *shape* of the structuring element. In this task the *size* of the structuring element is square i.e.  $2 \times 2$ ,  $3 \times 3$ , or  $4 \times 4$  etc. and can be of any shape.

The morphological functions use the following code to get the coordinates of the origin of structuring elements of any size and dimension.

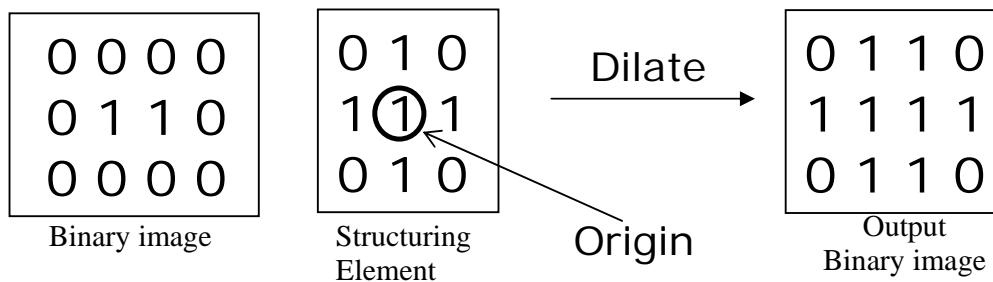
```
origin = floor(size(structuring_element)/2)
```

If structuring element matrix is  $[0 \ 1 \ 0; \ 1 \ 1 \ 1; \ 0 \ 1 \ 0]$

Then, `size(structuring_element)= 3×3`

So, `origin = (1,1)` [i.e. see the following figure]

Dilate (B,S) takes binary image B, places the origin of the structuring element S over each 1-pixel, and ORs the structuring element S into the output image at the corresponding position.



### Input

First line of the input file is an integer  $T$  ( $T < 25$ ) which denotes how many sets of inputs are there. Each test case starts with the dimensions of the binary image  $m \times n$  where  $m$  ( $2 \leq m \leq 100$ ) is the number of rows and  $n$  ( $2 \leq n \leq 100$ ) is the number of columns. Followed by the binary image which contains only 0 or 1. Then followed by the dimensions of the *structuring element*  $q \times r$  where  $q$  ( $1 \leq q \leq 10$ ) is the number of rows and  $r$  ( $1 \leq r \leq 10$ ) is the number of columns and then followed by the *structuring element*. *Size of structuring element* will be less than or equal to binary image ( $q \leq n, r \leq m$ ). **In input binary image border lines will not contain any 1's.**



## Output

For each test case print the binary image after dilation process.

*N.B: In output there will be no blank space (" ") after the end of a line.*

Sample Input	Output for Sample Input
2	Case 1:
3 4	0 1 1 0
0 0 0 0	1 1 1 1
0 1 1 0	0 1 1 0
0 0 0 0	Case 2:
3 3	1 1 1 0
0 1 0	1 1 1 1
1 1 1	1 1 1 1
0 1 0	0 1 1 1
4 4	
0 0 0 0	
0 1 0 0	
0 0 1 0	
0 0 0 0	
3 3	
1 1 1	
1 1 1	
1 1 1	

**Problem Setter : Md. Azher Uddin**

**Special Thanks : Tanveer Ahsan**

## Problem E: Little Rakin

Little Rakin is learning Math. While learning the Fibonacci series, he is talking over the phone to his paternal aunt who lives in New York. The Fibonacci series is:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} + F_{n-2}, \quad n > 1$$

But Rakin, who is listening to the visit that his uncle's family paid in the statue of Liberty on the phone, became very fascinated. He begins to imagine himself on the place, forgetting his present state. Due to his loss in concentration, what Rakin writes is:

$$F_0 = 0$$

$$F_1 = 1$$

$$F_n = F_{n-1} \times F_{n-2}, \quad n > 1$$

Now, you have to solve a problem which is similar to 'Little Rakin's Fibonacci series'. Here is a sequence given:

$$F_0 = a$$

$$F_1 = b$$

$$F_n = F_{n-1} \times F_{n-2}, \quad n > 1$$

Given **a**, **b**, and **n**, prime factorize  $F_n$ .

### Input

First line of the input will consist of the number of test cases  $T \leq 100$ . Then **T** cases follows.

For each case, there is a line containing three number **n**, **a**, **b**,

where :

$$2 \leq n \leq 40$$

$$2 \leq a, b \leq 1000$$

### Output

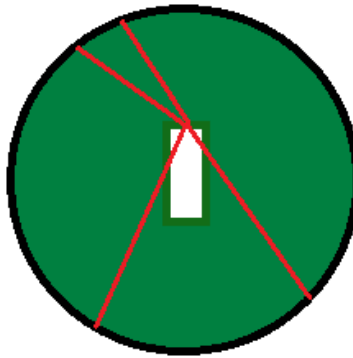
Print the prime factorization of  $F_n$  so that each prime number and its power will be in a single line separated by single space. If there is more than one prime, then prime numbers should be printed in increasing order. Print a blank line after each test case. For more clarification see the sample output format given below.

<b>Sample Input</b>	<b>Output for the Sample Input</b>
2	2 1
2 2 3	3 1
13 2 3	2 144 3 233

**Problem Setter : Mohammad Hafiz Uddin**  
**Alternate Solution : Md. Amjad Hossain Rahat**

## Problem F: Little Masters

Bangladesh whitewashed New Zealand in the three-match one day international series defeating the visitors by four wickets in the last and final match in Narayanganj. The Tigers scored 309 runs in 49.2 overs for the loss of six wickets at the Khan Shaheb Osman Ali Stadium in Fatullah. The most remarkable was the performance from the batsman of tiger side. Mominul, Nasir, Naeem, Mushfiq all showed tremendous batting performance. People call them little masters. They are clever enough to find a quick boundary. Here we have to do a little task for our busy little masters. We have to find out the length of the shortest and longest boundary distance from the batsman. We know the radius of the circular stadium and the position of the batsman. Center of the stadium is always the origin  $(0, 0)$ . **Boundary** means the perimeter of the circular field.



### Input

Input starts with an integer  $T$  ( $\leq 100$ ), denoting the number of test cases. Each case starts with a line containing three integers  $x, y, r$  ( $0 \leq x, y, r \leq 1000$ ).  $(x, y)$  denotes the coordinate of the batsman. And  $r$  denotes the radius of the stadium. You can safely assume that coordinate of the batsman will not be out of the stadium.

### Output

For each case, print the shortest and longest boundary distance. Show exactly 2 digits after decimal point, properly rounded. See the samples for exact formatting.

Sample Input	Output for Sample Input
3	100.00 100.00
0 0 100	5.00 15.00
3 4 10	0.00 10.00
5 0 5	

**Problem Setter : Kaysar Abdullah**

## Problem G: Breaking Board

Hector Salamanca, the cartel don aged before his years and is always confined to his wheelchair and oxygen tank. He never speaks a syllable. To express himself he used a board. The board was a 6\*6 2D grid as shown in the picture below. Top-left corner is (1, 1).

A	B	C	D	1	2
E	F	G	H	3	4
I	J	K	L	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z
5	6	7	8	9	0

To complete a sentence he goes character by character. For choosing a single character two steps involve:

1. Select the desired row of the character.
2. Select the desired column of the character.

Cost of choosing a character is the sum of row and column of the character in the board. Total cost of making a sentence is the sum of cost of choosing all characters. **You can assume that cost of choosing space of a sentence is 0.** For Example, cost of making sentence "CALL DEA" is  $(1+3) + (1+1) + (3+4) + (3+4) + (1+4) + (2+1) + (1+1) = 30$ .

In our problem Hector has a sentence to complete but the board is **not fixed**. We can break the board and reform it to minimize the cost of completing the sentence. We need to figure out what could be the minimal cost possible.

A	C	D	B	1	2
L	F	G	H	3	4
E	J	K	I	M	N
O	P	Q	R	S	T
U	V	W	X	Y	Z
5	6	7	8	9	0

This can be an optimal formation of board. Then the cost will be  $(1+2) + (1+1) + (2+1) + (2+1) + (1+3) + (3+1) + (1+1) = 21$ .

## Input

Input starts with an integer  $T$  ( $\leq 100$ ), denoting the number of test cases. Each case starts with a string of length  $L$  ( $\leq 100$ ) consisting of only uppercase letters (**A-Z**), digits (**0-9**) and spaces.

## Output

For each case, print the minimum possible cost in a single line. See the samples for exact formatting.

Sample Input	Output for Sample Input
2	21
CALL DEA	38
WALTER WHITE	12
09AZ	

**Problem Setter : Kaysar Abdullah**

**Alternate Solution : Prasanjit Barua**

## Problem H: Zero-Knowledge Protocol

In cryptography, a **zero-knowledge protocol** is a method by which one party (the *prover*) can prove to another party (the *verifier*) that a given statement is true, without conveying any additional information apart from the fact that the statement is indeed true. We can extend these ideas to verify whether anyone has properly find any pattern in a data stream without enclosing which position these pattern is found.

So, in this problem, you have to exactly match multiple pattern of length **M** in a number stream of length **N** and prove it in zero-knowledge protocol. A pattern **P** of length **M** will exactly match with a sub-pattern of stream **S** in **i<sup>th</sup>** index, if and only if ,

$$P[1...M] = S[i...i+M-1] , i+M-1 \leq N$$

In such case you have to sum **i<sup>2</sup>** in zero-knowledge protocol and verify it. You can assume that all the pattern indexing is **1-based**.

But, due to laziness and painfulness of generating multiple patterns, you have to use all the **distinct permutations** of given pattern **P** as multiple patterns.

### Input

First line of input will contain the number of test cases, **T ≤ 50** to follow. Each test case starts with a line given **N** and **M**. Then follows two lines. First line contains **N** numbers **s<sub>i</sub> (1 ≤ i ≤ N)** to denote number stream **S** and the next line contains **M** numbers **p<sub>i</sub> (1 ≤ i ≤ M)** to denote pattern **P**.

### Constraints

- $1 \leq N, M \leq 2 \cdot 10^4$
- $1 \leq s_i, p_i \leq 10^9$

### Output

For each case, print the total sum of squared match index in a single line. See the samples for exact formatting.

Sample Input	Output for Sample Input
2	5
3 2	5
10 11 10	
10 11	
4 3	
10 11 11 10	
10 11 11	

## Output Explanation

On the first test case two distinct permutation of P is possible. Pattern [10 11] is exactly matched with S in index 1 and pattern [11 10] is matched in index 2. So total sum in zero-knowledge protocol is  $1^2 + 2^2 = 5$ .

On the second test case three distinct permutation of P is possible. Pattern [10 11 11] is exactly matched with S in index 1 and pattern [11 11 10] is matched in index 2. Pattern [11 10 11] doesn't match anywhere. So total sum in zero-knowledge protocol is  $1^2 + 2^2 = 5$ .

**Problem Setter : Prasanjit Barua**

**Alternate Solution : Mohammad Hafiz Uddin**



## Problem I: Block Meh

Everybody knows Korimbai. Though he does not have time for any one. He is a legend in his own.

Now once Korimbai got very angry with the puny humans around him. He decided to block them on his social networking website Korimboi. ( Korimboi is very popular website among puny humans, every one of you have 1 or more account on that, though people do not know about the actual creator ( Korimbai ) / actual name ( Korimboi ) of the site, because as always Korimbai has no time for that.)

Korimbai is a strange man, so he wants to block them in a strange way. As the creator of korimboi he has this special power of blocking more than 1 people at 1 click. He does that in the following way.

He chooses one puny human given that puny human's entrance and exit time be  $S_1$  and  $E_1$  respectively in Korimadda (Chat client in Korimboi) and selects it for blocking. Then he finds another puny human whose entrance time and exit time be  $S_2$  and  $E_2$  respectively. He can select it if and only if  $S_2 > S_1$  and  $E_2 < E_1$ . Now he search for another puny human whose entrance time and exit time be  $S_3$  and  $E_3$  respectively and  $S_3 > S_2$  and  $E_3 < E_2$ . He continues this search until there are nobody available in those criteria that  $S_i > S_{i-1}$  and  $E_i < E_{i-1}$ . Then he blocks all of them at once in one click.

Now Korimbai wants to give this puzzle to you for your brain exercise (of course he knows the answer. He is THE Korimbai, but he has no time for that). He wants to know that given all the entrance and exit time for the puny humans in his korimadda list what is the minimum number of click Korimbai needs to give to delete all of the puny humans of his list.

### Input

First line of input will contain the number of test cases,  $T \leq 20$  to follow. In each test case 1<sup>st</sup> line contains  $N$ , number of puny humans in Korimbai's lists.  $N$  line follows each containing 2 integers  $S_i$  and  $E_i$ , entrance and exit time for the  $i^{\text{th}}$  puny human

$N \leq 20000$

$0 \leq S_i \leq E_i \leq 1000000$

### Output

For each input, print the output in the format, **Case X: Y** (here,  $X$  is the serial of the input and  $Y$  is the minimum number of click Korimbai needs to give to delete all of the puny humans of his list.

Sample Input	Output for Sample Input
2 4 1 5 2 5 3 5 3 4 4 1 5 2 4 3 5 3 3	Case 1: 3 Case 2: 2

### Output Explanation

On the first test case Korimbai can block 1<sup>st</sup> → 4<sup>th</sup> puny human in 1 click and 2<sup>nd</sup> in 1 click and 3<sup>rd</sup> in 1 click totaling 3 clicks. He cannot do less than that.

On the second test case Korimbai can block 1<sup>st</sup> → 2<sup>nd</sup> → 4<sup>th</sup> puny human in 1 click and 3<sup>rd</sup> in 1 click totaling 2 clicks. He cannot do less than that.

**Problem Setter : F. A. Rezaur Rahman**  
**Alternate Solution : Pratyai Mazumder**

## Problem J: GCD The Largest

Given  $N$ , print the largest number that can be achieved by taking gcd (greatest common divisor) of any two  $i$  and  $j$  where  $i \neq j$  and  $1 \leq i, j \leq N$ .

### Input

First line of input will contain the number of test cases,  $T \leq 2000$ . Then  $T$  cases follow. For each case, there is a line containing one integer  $N$  where  $2 \leq N \leq 10^{18}$ .

### Output

For each case, print one line containing a single integer which is the largest gcd of all pairs of numbers between  $1$  to  $N$ .

Sample Input	Output for Sample Input
2	1
2	2
5	

### Output Explanation

In the second case the GCD table is:

	1	2	3	4	5
1	-	-	-	-	-
2	1	-	-	-	-
3	1	1	-	-	-
4	1	<u>2</u>	1	-	-
5	1	1	1	1	-

Here the largest gcd of all pairs of numbers between  $1$  to  $5$  is  $2$ .

**Problem Setter : Muhammad Ridowan**