DIU TAKE OFF Programming Contest
Summer 2017, Permanent Campus

Organized By,
Computer & Programming Club
Daffodil International University
Permanent Campus

A. Speed & Accuracy

Score: 1

CPU: 1s
Memory: 512MB

Today is **“DIU Take Off Programming Contest – Summer 2017, Permanent Campus”** and the contest is running in DIU Permanent Campus. As you are a good programmer, this is the time to show others that how fast and accurate you are. Let’s see!

Input:

There is no input in this problem.

Output:

Welcome to Daffodil International University, Permanent Campus

B. First Date

Score: 1

CPU: 1s
Memory: 512MB

Gazi, a student of Tourism and Hospitality Management (THM) got introduced himself with **Mou**, a student of Computer Science and Engineering (CSE) department on Facebook. They like each other very much but never met yet formally. Since DIU has a very large campus, departments are far away. So, they decided to meet each other today. As both of them are very smart, they will try to fix a place for meeting so that they can cover same distance of the total path. Now as a programmer, your task is to help them to find the place where they will meet to gain their purpose.

Input:

Input starts with two integers A and B `(0 <= A <= B < 1000)` denoting the position of both **Gazi** and **Mou** respectively.

Output:

Print the integer position of the place where they will meet.

Samples

Input	Output
10 20	15

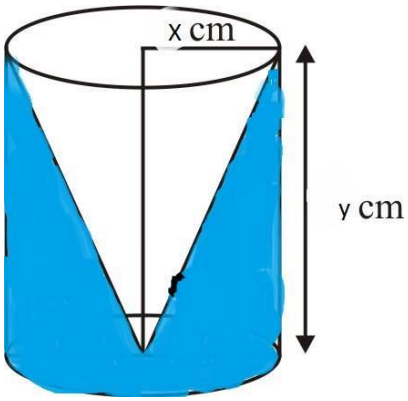
Input	Output
20 35	27

C. A simple geometry

Score: 1

CPU: 1s
Memory: 512MB

Lubna, a student of DIU pharmacy, entered into the lab and saw that a cone and a cylinder both having equal radii and heights. The cylinder had its one side open and it was full of water. Then she inserted the cone into the cylinder and the cone tightly fitted the cylinder **(follow the picture)**. In effect, the water overflowed the upper level of the cylinder. Now you have to calculate how much water remained in the cylinder.



Blue highlighted portion indicates the remaining water.

Input

The first line of input is followed by an integer `T (1 <= T <= 200)`. Next `T` lines contain two integer `x (1 <= x <= 1000)` and `y (1 <= y <= 1000)` indicating the **radii** and **heights** of cone and cylinder.

Output

For each test case you have to print how much water remained in the cylinder in a floating number with `Three digits` after decimal point. (Warning about big output)

Sample

Input	Output
3	150.796
6 2	821.002
7 8	339.292
9 2	

We all know that, volume of a cylinder is $\pi r^2 h$ and here the volume of cone is `one third` of the volume of the cylinder. The value of π is `3.14159`.

Problem Setter (C. A simple geometry):
Fahim Abrar
161-15-903
ACM Team
DIU CPC-PC

D. Legend Hablu

Score: 1

CPU: 1s
Memory: 512MB

Mr. Hablu is the greatest legend in DIU Permanent Campus for his programming and problem-solving skill. No one knows who this legend is except ACM Team DIU CPC-PC. He always hides himself. As you are a good programmer, you are very curious to meet him. So, you got some clues from ACM Team how to meet the legend. You got a position of Mr. Hablu but you can't directly meet him in that place. You have to use some teleportation machine to reach there which was created and set by Mr. Hablu.

The first machine is in **AB-01** building which will teleport you to **AB-03** building. There will be next machine in **AB-03** building which will teleport you to **Auditorium**. And from **Auditorium**, the final machines will teleport you to **AB-04** Building where Mr. Hablu is waiting for you. But there is some teleportation cost. This is not a usual paying rule. Suppose you have `N` taka. Then you have to pay `N-{(N/2)-1}` Taka from your `N` Taka for one teleportation charge. Then for next teleportation charge, again you have to pay with the same rule from your remaining money and so on. You have to ignore floating point values while paying. That means only integer values will be counted. But Mr. Hablu set a tricky part here. You have to take `N` Taka so that after you reach your destination, you will only have `1` Taka remaining. Otherwise you will never meet him again. Now you have to find for which value of `N`, your final remaining money will be `1` Taka after you reach your destination **(AB-04)**?

Input:

Input starts with an integer `T (≤ 500)`, denoting the number of test cases in first line of input. Each test case will have one number, which denotes `N (1 <= N <= 100000000)`

Output:

If the value of N fulfills the above condition, then print **"Greetings from Hablu"** (without quotes). Otherwise print **"Goodbye"** without quotes.

Sample

Input	Output
2	Goodbye
500	Greetings from Hablu
23	

Clarification: on the first test case, you have **500** Taka. When you will reach your destination, your remaining money will be **60** Taka which doesn't satisfy the above condition discussed in the problem description.

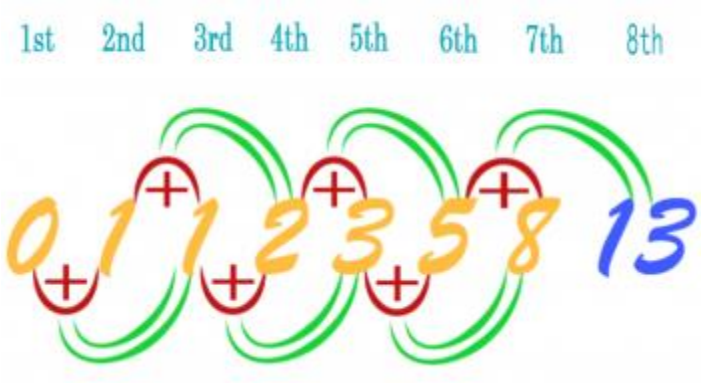
Problem Setter (D. Legend Hablu):
Rijoanul Hasan Shanto
152-15-572
ACM Team
DIU CPC-PC

E. Fun with Fibonacci

Score: 1

CPU: 1s
Memory: 512MB

The **Fibonacci sequence** is a set of numbers that starts with a **0**, followed by a **1**, and proceeds based on the rule that, each number ***(called a Fibonacci number)*** is equal to the sum of the preceding two numbers. If the Fibonacci sequence is denoted by ***F (n)***, Then the n^{th} term will be ***F (n) = F (n-1) + F (n-2)***. And some first numbers of the sequence are **0, 1, 1, 2, 3, 5.....**etc.



Now your task is to find the number of `odd` and number of `even` number(s) between any given range.

Input:

First Line of the input contains **T**, representing the number of test cases $(1<=T<=50)$. Each test case contains two integers **N** and **M** $(1<=N<=M<=10^{18})$ and $(|N-M|<=10^{5})$, where **N** is the N^{th} Fibonacci number and **M** is the M^{th} Fibonacci number of the sequence. Now, you have to calculate the number of `odd` number(s) and number of `even` number(s) between **N** and **M**(*inclusive*).

Output:

For each test case print case number with the desired result as shown in the sample output.

Sample

Input	Output
2 2 6 1 5	Case 1: Odd = 4 Even = 1 Case 2: Odd = 3 Even = 2

Problem Setter (E. Fun with Fibonacci):
Md. Razibul Hasan Mithu
161-15-882
ACM Team
DIU CPC-PC

F. Biggest Power

Score: 1

CPU: 1s
Memory: 512MB

Fahim, an idle programmer who loves to solve very easy mathematical problems. Currently, he is solving a problem on finding out the `last digit` of a number. This is a very easy task. But in this problem, you are given a more complicated task. You will be given a number `n`. You have to find out the last digit of 9^n.

Input

The first line of input contains an integer `T (1 <= T <= 100)`, denoting the number of test cases. Each test case consists of one line containing an integer `n (0 <= n <= 10^16).`

Output

You have to print the desired result as shown in sample output.

Sample

Input	Output
1 1	Case 1: 9

Problem Setter (F. Biggest Power):

Amir Sohel
161-15-874
ACM Team
DIU CPC-PC

G. Alien in DIU, Permanent Campus

Score: 1

CPU: 1s
Memory: 512MB

Today is **DIU Take Off Programming Contest**. Every single person around the world knows about this contest. Even outside the world, there exists a planet named ***"Programming World"***. **DIU CPC-PC** invited the aliens of ***Programming World*** to attend the contest. It is an interesting fact that, those aliens look like `English characters`. In order to attend the contest, aliens landed at the **Tejgoan Airport**and they are now currently waiting at different bus stands for the bus provided by **DIU** *permanent campus* to transport them to the venue. But the bus seats are not enough to transport all the aliens. So the bus driver asks for an algorithm to the **CPC,Permanent Campus** to avoid this limitation. So a team of best programmers of **CPC** proposed an algorithm and the algorithm is something like, the bus driver will go to the `1st` bus stand and pick up the alien who is in the `1st` position, then he goes to the `2nd` bus stand and pick up the alien standing in the `2nd` position, then goes to the `3rd` bus stand and pick up the one in the `3rd` position of the aliens group and continues for the rest of the bus stand in the same fashion. It is worth mentioning that at every bus stand, there is some unknown number of aliens waiting and all bus stands are in the same straight line. Now as a programmer of **DIU** *permanent* campus, your task is to write a program to help the bus driver so that he can decide on which `participants` will get a ride on the bus.

Input:

Input starts with an integer `T (<=30)`, denoting the number of test cases in first line of input. Each test case starts with an integer `N` represents the total number of alien's groups `(0<N<=200)` standing in different bus stands. Total number of aliens will never exceed the limit **10^3.**

Output:

For each test case, you must print which aliens can get into the bus for contest in the same format as given in the sample output.

Sample

Input	Output
2	

Input	Output
3 Hey Good Lawyer 5 Are you a good programmer	

Problem Setter (G. Alien in DIU, Permanent Campus):
Mohammad Shakil Mahmud
161-15-884
ACM Team
DIU CPC-PC

H. Friends Forever (But No Sacrifice with Food)

Score: 1

CPU: 1s
Memory: 512MB

We all know about the Inter University Cricket tournament. This tournament occurred in every year. **Daffodil International University**participated in this tournament previous years. In the running tournament **DIU** and **NSU** will participate, So **Hasan** and **Gazi** wants to enjoy this match from the stadium gallery. But the problem is in the stadium that one person can carry *single* food item. So, Hasan wants to carry ***popcorn*** and **Gazi** wants to carry ***Coca-Cola***. Unfortunately, they can't sit *alongside*. But they want to share *popcorn* and *Coca-Cola* with each other. They can use many ways for their sharings.

As they are aware about time, so they want to minimize `time`. At least how many chairs do they need to pass in order to meet each other?

As you are a good programmer, you can write a program that will help **Hasan** and **Gazi** to find their `shortest distance` *to meet and share their foods*.

Input:

The First line of the input consists of one Integer `T` `(1<=T<=20)` denoting the number of test cases. Next line of the input consists of two integers `R` `(1<=R<=10000)` and `C` `(1<=C<=10000)`, denoting the number of ***row*** and number of ***column*** of the Stands of this stadium. Next **R** row and **C** column consist of uppercase English letter **'C'**,**'H'**,**'G'** here **'H'** and **'G'** denoting **Hasan's** and **Gazi's** chairs respectively and **'C'** denotes other chairs.

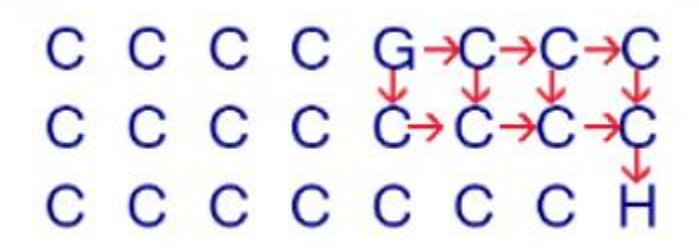
Output:

You have to calculate `minimum` number of chairs they need to pass. Print the case number and desired output as sample output.

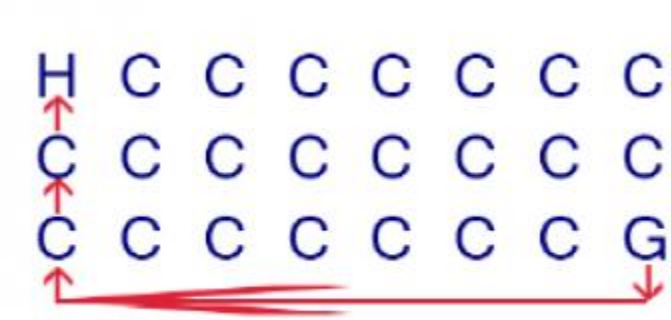
Sample

Input	Output
2 3 8 CCCCGCCC CCCCCCCC CCCCCCCH 3 8 HCCCCCCC CCCCCCCC CCCCCCCG	Case 1: 5 Case 2: 3

Output clarification:



In the first example they can use any of these routes, the output is `5`.



In the `2nd` example they can use this way for minimization `time`, because the stadium is `round` figure so *`left and right`* side is adjacent. So, the output is `3`.

Problem Setter (H. Friends Forever (But No Sacrifice with Food)):

Md. Razibul Hasan Mithu
161-15-882
ACM Team
DIU CPC-PC