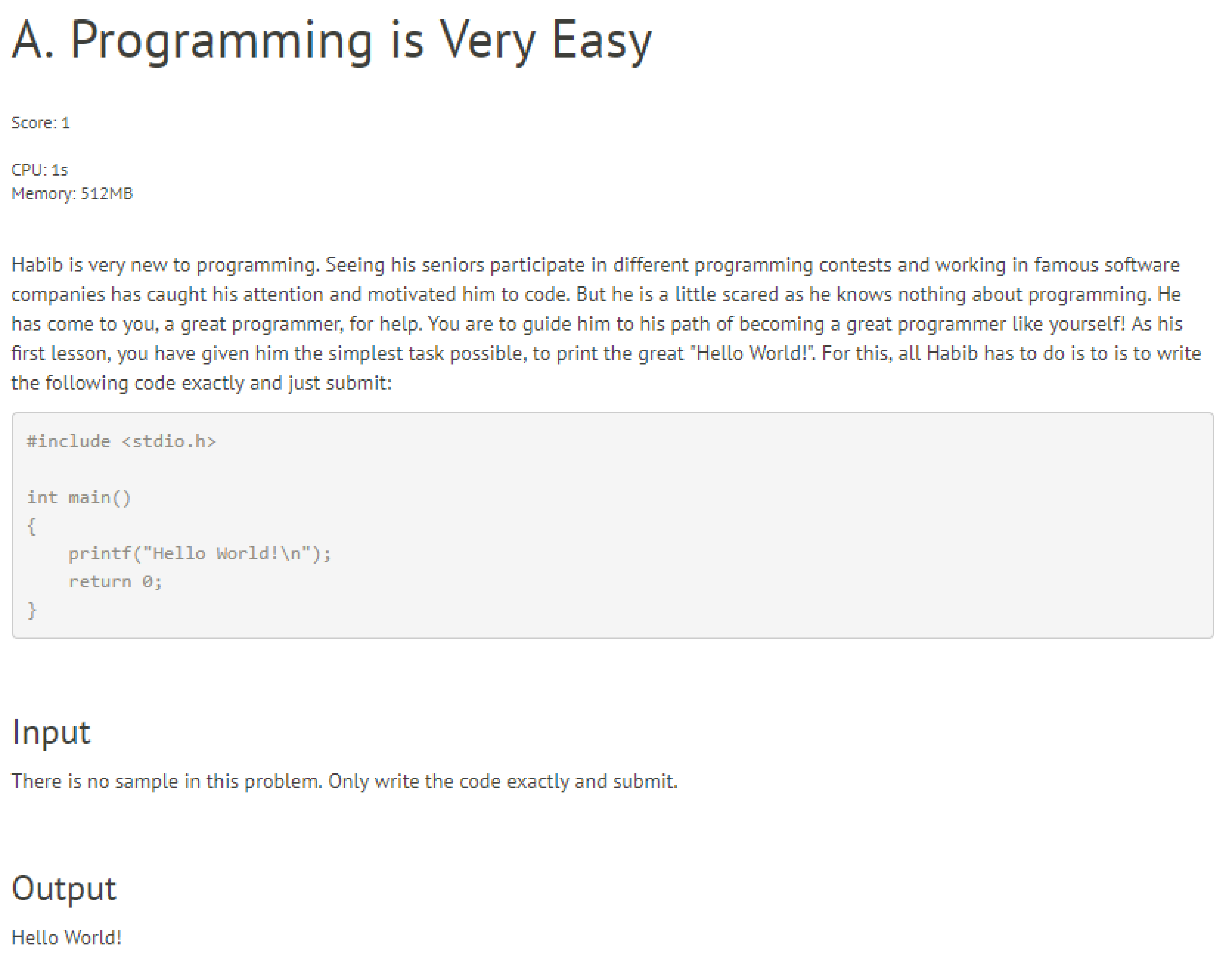


A. Programming is Very Easy

Score: 1

CPU: 1s
Memory: 512MB

Habib is very new to programming. Seeing his seniors participate in different programming contests and working in famous software companies has caught his attention and motivated him to code. But he is a little scared as he knows nothing about programming. He has come to you, a great programmer, for help. You are to guide him to his path of becoming a great programmer like yourself! As his first lesson, you have given him the simplest task possible, to print the great "Hello World!". For this, all Habib has to do is to is to write the following code exactly and just submit:

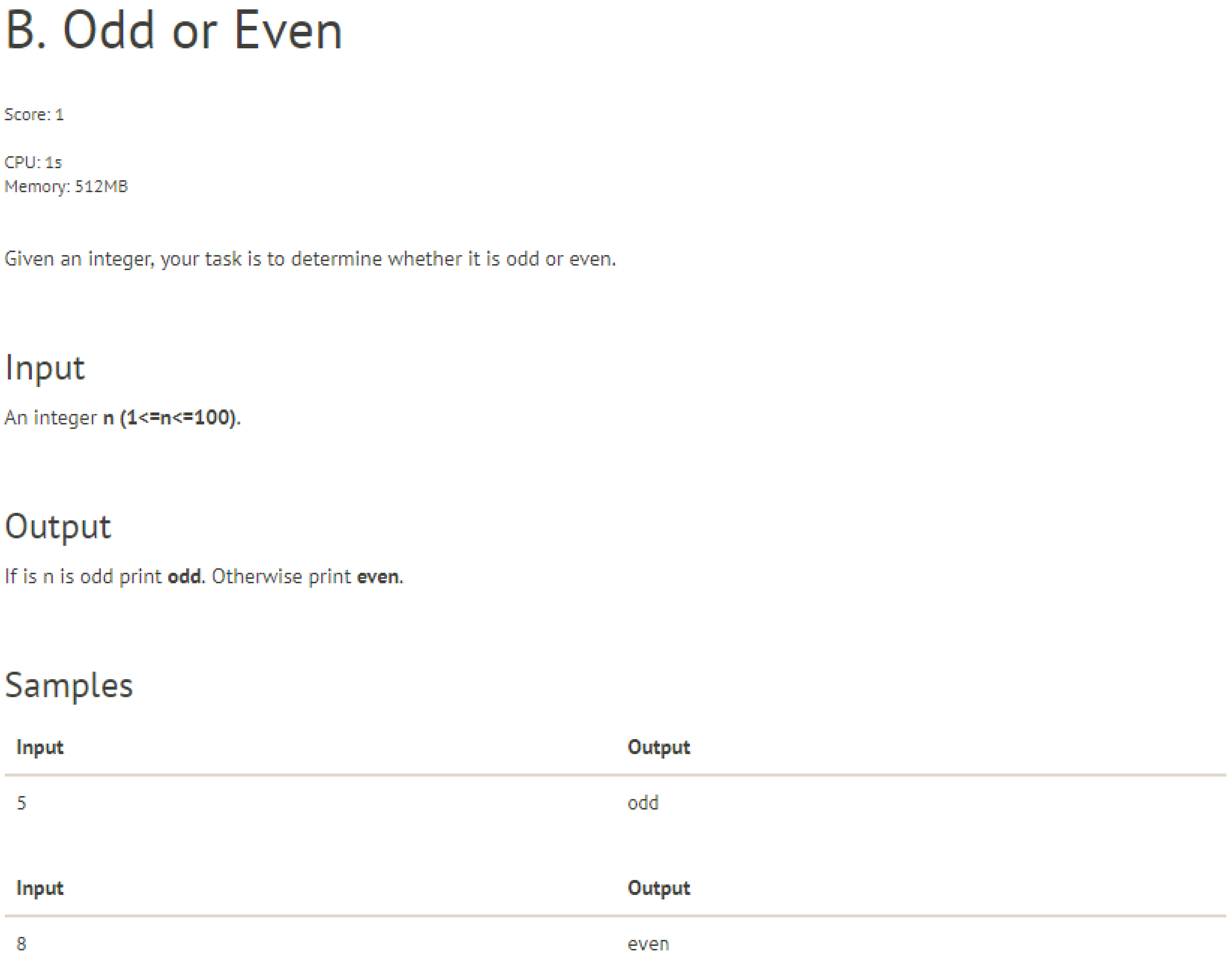
```
#include <stdio.h>

int main()
{
    printf("Hello World!\n");
    return 0;
}
```

Input

There is no sample in this problem. Only write the code exactly and submit.

Output

Hello World!



B. Odd or Even

Score: 1

CPU: 1s
Memory: 512MB

Given an integer, your task is to determine whether it is odd or even.

Input

An integer **n (1<=n<=100)**.

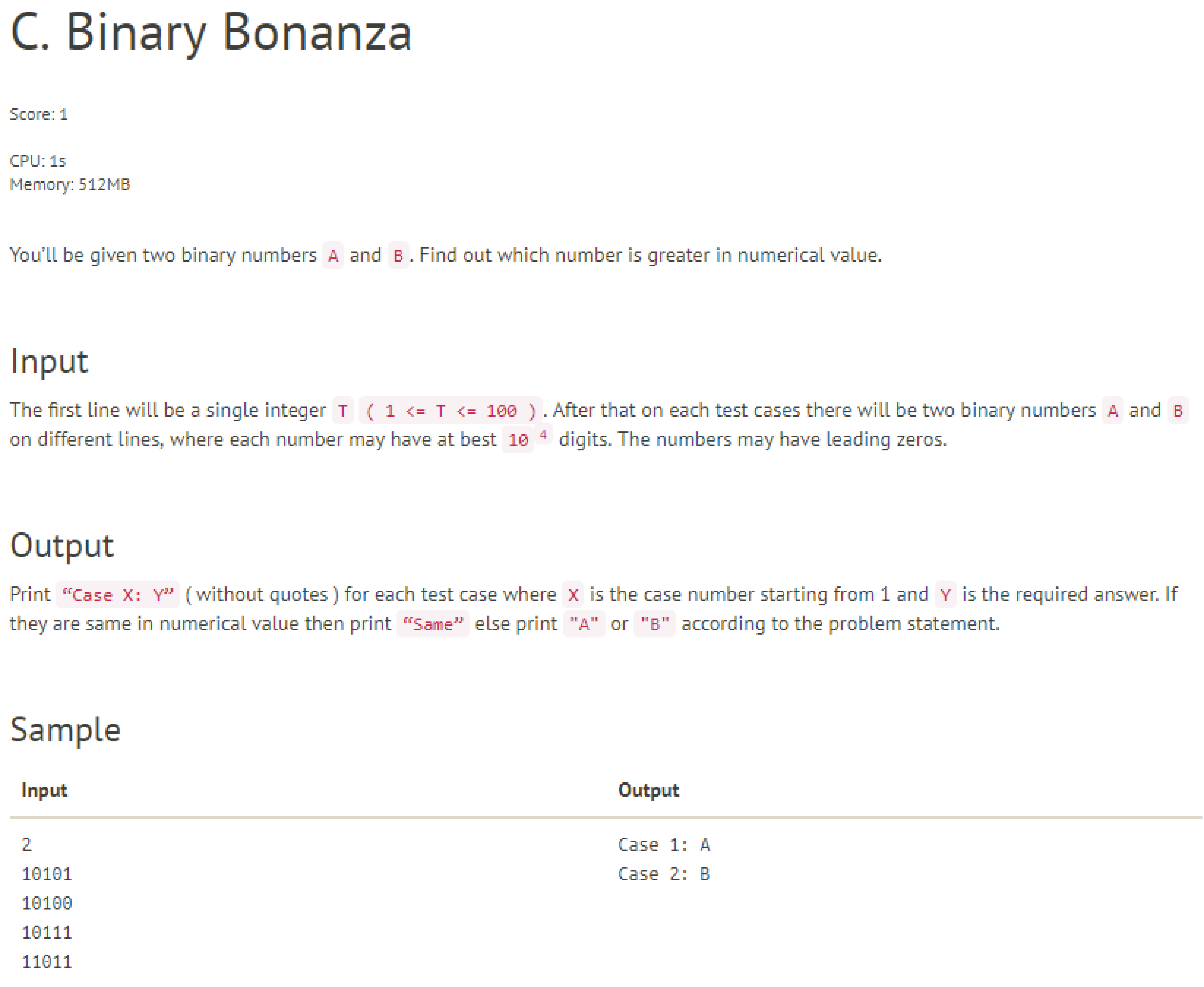
Output

If is n is odd print **odd**. Otherwise print **even**.

Samples

Input	Output
5	odd

Input	Output
8	even



C. Binary Bonanza

Score: 1

CPU: 1s
Memory: 512MB

You'll be given two binary numbers `A` and `B`. Find out which number is greater in numerical value.

Input

The first line will be a single integer `T` `( 1 <= T <= 100 )`. After that on each test cases there will be two binary numbers `A` and `B` on different lines, where each number may have at best 10^4 digits. The numbers may have leading zeros.

Output

Print `“Case X: Y”` (without quotes) for each test case where `X` is the case number starting from 1 and `Y` is the required answer. If they are same in numerical value then print `“Same”` else print `"A"` or `"B"` according to the problem statement.

Sample

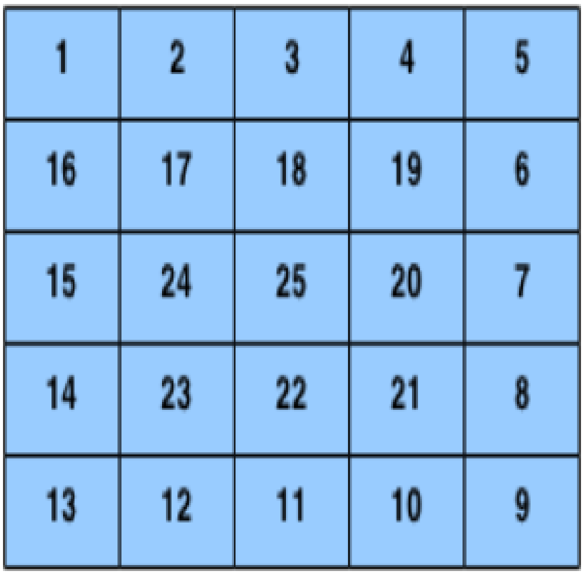
Input	Output
2 10101 10100 10111 11011	Case 1: A Case 2: B

D. Spiral Matrix

Score: 1

CPU: 1s
Memory: 64MB

Today is Rajdip's birthday. I know he likes spiral matrix very much. So I want to give him a spiral matrix . I went to many shop but didn't find a spiral matrix there. I am not able to make a spiral matrix. Then I heard that you are a very good programmer so I need your help. Can you make a spiral matrix for me? An example of a 5x5 spiral matrix is given below:



1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Input

The first line contains a integer `T` (`1<=T<=25`) The number of test cases. Next `T` lines contains integer `N` (`1<= N<=25`) - for each N you have to make a spiral matrix.

Output

For every N you have to make a NxN spiral matrix. For every number in the spiral matrix box you have to print like `printf("%4d", your_number_for_this_cell)` which prints an integer in a 4 digit place.

Print a blank line after each test case.

Sample

Input

```
1
5
```

Output

```
   1   2   3   4   5
  16  17  18  19   6
  15  24  25  20   7
  14  23  22  21   8
  13  12  11  10   9
```

E. Prime Game

Score: 1

CPU: 1s
Memory: 512MB

"A prime number is a natural number which has exactly two distinct natural number divisors" in other words, "A prime number is a number that has no proper factors". First few prime numbers are: 2, 3, 5, 7, 11, 13, ... and so on.

Your task is to write a program that reports. Given two integers `N` and `M`, is the sum of `N`'th prime and `M`'th prime even or odd?

Input

The input file contains multiple cases. Each test case will consist of a line containing two integers `N` and `M` where `( 1 <= N, M <= 10000000 )`. The input is terminated by a case where `N` and `M` both are equal to 0 (ZERO), this case should not be processed.

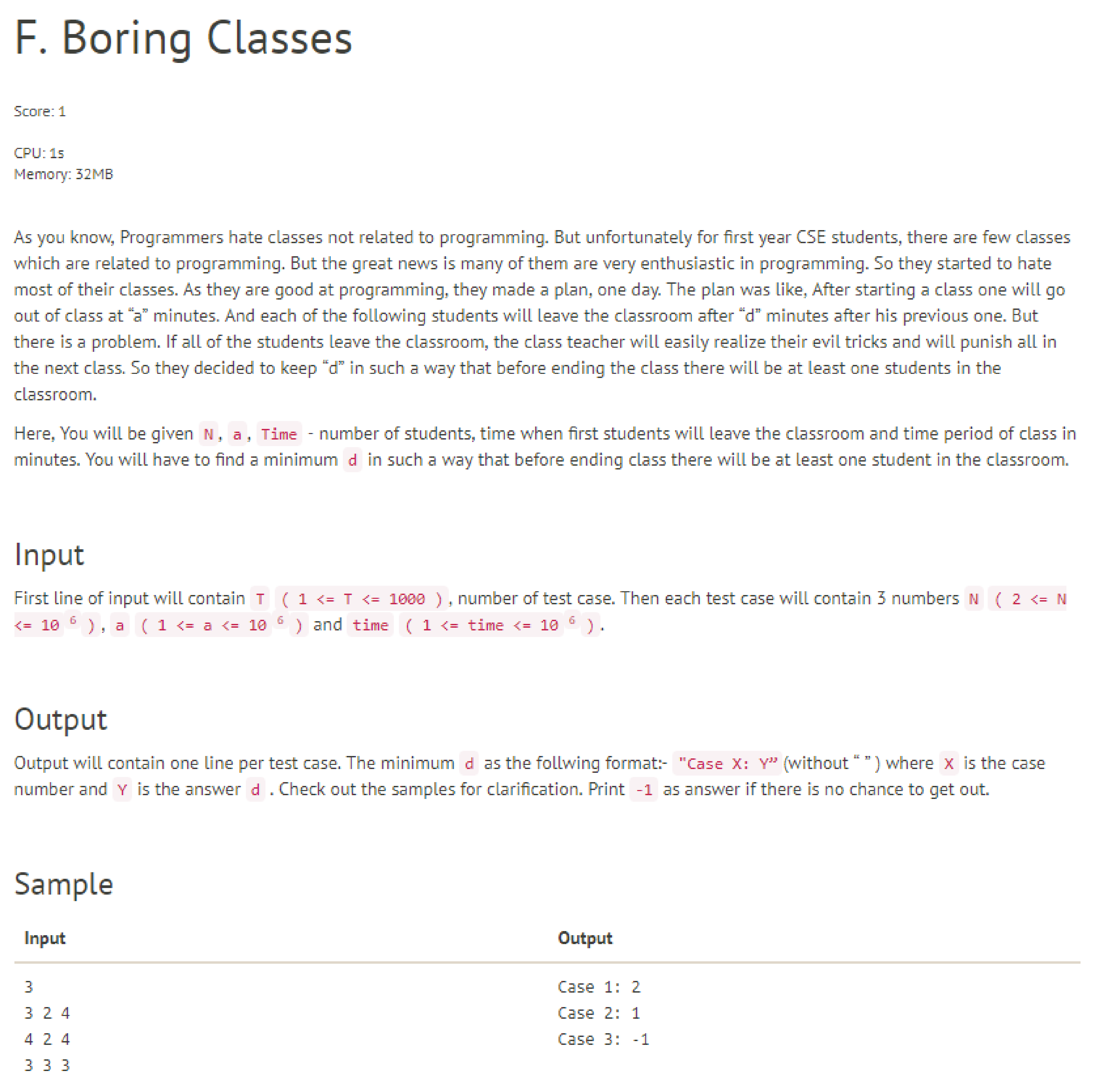
Output

For each case print `"Case X: Y."` without quotes where `X` is the case number(starting from 1) and `Y` is a single word `"Even"` or `"Odd"` (without quotation) as per description.

Sample

Input	Output

2 4
1 2
0 0
	Case 1: Even.
	Case 2: Odd.



F. Boring Classes

Score: 1

CPU: 1s
Memory: 32MB

As you know, Programmers hate classes not related to programming. But unfortunately for first year CSE students, there are few classes which are related to programming. But the great news is many of them are very enthusiastic in programming. So they started to hate most of their classes. As they are good at programming, they made a plan, one day. The plan was like, After starting a class one will go out of class at “a” minutes. And each of the following students will leave the classroom after “d” minutes after his previous one. But there is a problem. If all of the students leave the classroom, the class teacher will easily realize their evil tricks and will punish all in the next class. So they decided to keep “d” in such a way that before ending the class there will be at least one students in the classroom.

Here, You will be given `N`, `a`, `Time` - number of students, time when first students will leave the classroom and time period of class in minutes. You will have to find a minimum `d` in such a way that before ending class there will be at least one student in the classroom.

Input

First line of input will contain `T` `( 1 <= T <= 1000 )`, number of test case. Then each test case will contain 3 numbers `N` (2 <= N <= 10^{6}), `a` (1 <= a <= 10^{6}) and `time` (1 <= time <= 10^{6}).

Output

Output will contain one line per test case. The minimum `d` as the follwing format:- `"Case X: Y”`(without “ ”) where `X` is the case number and `Y` is the answer `d` . Check out the samples for clarification. Print `-1` as answer if there is no chance to get out.

Sample

Input	Output

3
3 2 4
4 2 4
3 3 3
	Case 1: 2
	Case 2: 1
	Case 3: -1

G. Arrange the Teams

Score: 1

CPU: 3s
Memory: 512MB

Mr. Mamun has been given the responsibility of arranging this year's BronqueDelowahRoff Cup (BDR-Cup). Teams from over a hundred different institutions are participating this year, and Mr. Mamun has been asked to create the seating arrangement for the teams. But Mr. Mamun does not trust the participating teams and believes that if two teams from the same institution are seated together, they are more likely to cheat. So he wants to arrange the seats such that no two teams from the same institution are seated together.

Now, help Mr. Mamun find the number of ways he can assign seats to the teams such that no two teams from the same institution are seated beside each other in any row.

More formally, Given the number of rows `R`, the number of columns `C`, and the number of different institutions `K` Find an arrangement **a** where,

$a_{i,j} \neq a_{i,j+1}$; for all 1 <= i <= R and 1 <= j <= C

where, $a_{i,j}$ is the institution of the team assigned to the seat at the i^{th} row of the j^{th} column.

Input

First line contains `T`, the number of test cases. Next `T` lines contain 3 Integers `R`, `C` and `K` Where, `R` is the Number of Rows, `C` is the Number of Columns and `K` is the Number of different institutions.

Output

For each test case, output the answer in the format, `Case x: y` Where, `x` is the number of the test case and `y` is the answer for that test case. **Since the number may be quite large, output the answer modulo 10^9+7.**

Constraints

T <= 100
1 <= R, C, K <= 10^5

Sample

Input	Output

4
3 3 2
2 4 2
3 5 3
100 100 100
	Case 1: 8
	Case 2: 4
	Case 3: 110592
	Case 4: 708857890

Note: Mr. Mamun is a very heartless person, so some institutions may not get any seats but he does not care.

H. Longest Decreasing Subsequence (LDS)

Score: 1

CPU: 1s
Memory: 512MB

Suppose you have an array `A` = `{ 7, 3, 6, 8, 2, 9 }`. Now if we write down some decreasing subsequences of array `A`, then we will find

`8, 7, 2`
`7, 6, 2`
`3, 2`
`6, 2`
`8, 2`

Among all the decreasing subsequences the longest decreasing Subsequence is `7, 8, 2` and `7, 6, 2`.
So the length of the `Longest Decreasing Subsequence` is `3`.

You will be given the array size and the elements of the array. You have to find the `length` of the longest decreasing sub sequence of that array.

Input
First you will be given a positive number `T(T<=100)`, number of test case. For each test case you will be given the size of the array `N(1<N<1000)`. Then in next lines you will be given `N` space separated integers. No number of the array will be greater than `100000`.

Output
For each test case you have print one positive integer `L`, which is the length of the longest decreasing subsequence of the given array.

Sample

Input	Output

4
6
7 3 6 8 2 9
2
1 2
3
2 3 1
5
1 3 2 4 5
	3
	1
	2
	2