UTD Programming Contest for High School Students
April 7th, 2018

- **Time Allowed: three hours.**
- **Each team must use only one computer - one of UTD's in the main lab.**
- **Answer the questions in any order.**
- **Use only Java, C, C++, or Python 2 or Python 3**
- **Your program source code must be contained in one source file.**
- **Do not use the "package" construct in your Java code.**
- **Your programs must read from the named file specified in the problem header, and must output to System.out (cout for C/C++ programmers.)**
- **Do not access the web except to access Java C/C++, or Python documentation.**
- **Do not use any recording device containing code, other than UTD's computer.**
- **Your solutions must be entirely yours, typed by you during the time allowed for the contest.**
- **As soon as you have solved a problem, submit ONLY the source file via your `PC^2` client. Don't wait for the judges' response, keep going!**

Scoring

Equal points will be awarded to each question, even though they may not be equally difficult.

In order to break ties, we will use penalty points. The penalty points for a question will be zero if the question is not answered correctly. If a correct submission occurs for a question at time T minutes, then T penalty points will be added, plus 20 penalty points for each incorrect submission for that question.

The top ranked three teams, irrespective of category (Novice or Advanced) will be deemed 1st, 2nd and 3rd place-winners of the Advanced Contest. This rule in effect promotes any Novice teams to Advanced Status. Those teams cannot also be place-winners in the Novice Contest.

Of the remaining Novice teams, the top three ranked teams will be place-winners of the Novice category.

A. What Started This?

Input File: A.txt
Runtime allowed = 10 seconds

Given a recurrence relation, such as $x_i = 3x_{i-1} - 5$ and the knowledge that $x_5 = 1825$, what was x_0?

If, say $x_0 = 10$, then $x_1 = 3 \times 10 - 5 = 25$ and $x_2 = 3 \times 25 - 5 = 70$ and so on.

The recurrence relations that you must work with will all be of the form

$$x_i = A \times x_{i-1} + B$$

Input
The first line of input contains a single decimal integer P, $(1 \leq P \leq 10000)$, which is the number of datasets that follow. Each dataset is made up of one line of text. The first value on each line is the problem number, starting at 1. This is followed by the integers A and B. Then comes k, the subscript of x at which the value, x_k is given.The corresponding value of x_k is an integer and is the last entry on the problem line.

There will be no more than 20 test cases and in each test case $A, B \in \{-20, 20\}$, $k \in \{2, 20\}$

The x_i values will be integers, $-2^{32} \leq x_0, x_k \leq 2^{32}$

Output
The output has one line for each dataset giving the dataset number, as illustrated below, followed by a single space character, and then the integer value of x_0.

Sample Input	Output for Sample Input
2 1 3 -5 5 1825 2 2 2 6 254	1 10 2 2

B. Chanukah

Input File: B.txt
Runtime allowed = 10 seconds

The Jewish holiday of Chanukah lasts for eight days and eight nights. On the evening before each day, candles are lit in a menorah. On the first evening, one candle is lit, on the second, two are lit, and so on. However, each evening, an extra candle, called the Shammas, is also lit (in fact, this candle is used to light the other candles). Thus, for the entire holiday, 44 candles are necessary. But what if Chanukah lasted a different number of days? How many candles would be needed? For this problem, you will write a program that determines how many candles would be necessary for a Chanukah holiday lasting for a given number of days.

Input

The first line of input contains a single decimal integer P, $(1 \leq P \leq 10000)$, which is the number of datasets that follow.

Each dataset should be processed identically and independently. Each dataset consists of a single line of input. It contains the dataset number, K, followed by a single space, followed by a single decimal integer N, $(1 \leq N \leq 10000)$, which gives the number of days to assume for the holiday.

Output

For each dataset there is one line of output. The single output line consists of the dataset number, K, followed by a space followed by the number of candles needed for an N-day Chanukah holiday.

Sample Input	Output for Sample Input
3	1 44
1 8	2 2
2 1	3 65
3 10	

C. Sum Squared Digits

Input File: C.txt
Runtime allowed = 10 seconds

To form the *Sum Squared Digits function*, $SSD(b, n)$ of a positive integer n, in base b, we first state that the digits of n, $a_0, a_1, \ldots, a_{k-1}$, are in the set $\{0, 1, \ldots, b-1\}$. Then the *Value* of n is given by

$$Value(n) = a_{k-1}b^{k-1} + a_{k-2}b^{k-2} + \ldots + a_1b^1 + a_0$$

Then

$$SSD(b, n) = a_0^2 + a_1^2 + a_2^2 + \ldots a_{k-1}^2$$

is the sum of squares of the digits of the representation.

Write a program to compute the *Sum Squared Digits Function* of an input positive number.

Input

The first line of input contains a single decimal integer P, $(1 \leq P \leq 10000)$, which is the number of datasets that follow. Each dataset should be processed identically and independently.

Each dataset consists of a single line of input. It contains the dataset number, K, followed by the base, b $(3 \leq b \leq 16)$ as a decimal integer, followed by the positive integer, n (as a decimal integer) for which the Sum Squared Digits Function is to be computed with respect to the base b. The value of n will fit in a 32 bit unsigned integer.

Output

For each dataset there is a single line of output.

The single line of output consists of the dataset number, K, followed by a single space followed by the value of $SSD(b, n)$ as a decimal integer.

Sample Input	Output for Sample Input
3 1 10 1234 2 3 98765 3 16 987654321	1 30 2 19 3 696

D. Less Than Me?

Input File: D.txt
Runtime allowed = 30 seconds

Given an array of N integers and Q queries, Q $(1 < Q < N < 10^6)$, determine how many array elements are less than and to the left of the value at each query index.

For example, if the array contains integers

Index	0	1	2	3	4	5	6	7	8	9	10
Content	3	5	2	8	6	9	11	1	7	6	3

The query 4 will give the result 3 since the first four values in the array contain three values less than 6. The query 8 would give the result 5.

You may assume that elements in the array are randomly distributed although many duplicates may exist.

Input:
The first line of input contains a single decimal integer P, $(1 \leq P \leq 30)$, which is the number of datasets that follow. Each dataset should be processed identically and independently.

Each dataset comprises three lines of text. The first line consists of two space-separated integers: N, the number of elements in the array, and Q, the number of queries. The second line contains N space separated integers representing the values in the Array. Each value is in $[-10^9, 10^9]$. Those values are randomly distributed. The third line contains Q space-separated integer queries, each one in $[1, N-1]$.

Output:
For each dataset output Q lines of text, each one giving the query number, starting at 1, then a space, and then the result of the query.

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

E Uncle Rohi

Data File: E.txt
Runtime allowed = 10 seconds

While locked away in prison, old uncle Rohi decides to get in shape. While doing crunches, push-ups, and other favorties, Rohi gets bored. To spice things up, he defines a few fun rules to amend his counting style.

- He will pick some integer X to start counting from.
- He will pick some integer N to count by.
- When he gets to the number 2147483647, his count overflows, down to -2147483648.
- He follows the rules for basic 32 bit signed integer overflow.
- He will stop his exercise when his count gets to 3.
- He is wise enough to pick only numbers that will allow him to stop.

As a guard, you get bored too, so you decide to figure out how many repetitions of each exercise Rohi will do with given values for X and N.

Input:
You will be given an unknown number of datasets, each only one line long. Each line consists of the problem number P, $(1 \leq P \leq 100)$, followed by two integers, X and N, the two values that Rohi is using.

Output:
For each dataset, output one line beginning with the problem number P, then a space character, then the number of repetitions Rohi makes.

Sample Input	Output for Sample Input
1 7 9	1 477218588
2 60 31	2 3048041305
3 10 11	3 3904515723
4 5 10	4 858993459
5 53 6	5 1431655757

F. Don't Eat The Dead Fish

Input File: F.txt
Runtime allowed = 10 seconds

Two perfect logicians, A and B, play a game. It begins with a fish bowl containing some number, N, of M&Ms and a dead fish. There is no water in the bowl. That's how the fish died. The two players alternate turns. In one turn a player must remove some number in $[1, 2, \ldots, M]$ M&Ms. The object of the game is to leave your opponent with the dead fish.

Given N and $M, (2 \leq M < N \leq 1000)$, and the name of the starting player (either A or B), you only have to decide who wins.

Input:

The first line of input contains a single decimal integer $P, (1 \leq P \leq 500)$, which is the number of datasets that follow. Each dataset should be processed identically and independently. Each dataset consists one line of input containing the problem number K, then two integers N and M and the character A or B. The four fields are separated by single spaces.

Output:

For each dataset there is one line of output containing the problem number, then a space, then the name of the winning player.

Sample Input	Output for Sample Input
2 1 10 4 A 2 8 3 B	1 B 2 A

G. Almost Pisano

Input File: G.txt
Runtime allowed = 10 seconds

You are familiar with the Fibonacci sequence, 1,1,2,3,5,8,13, ... etc. where

$$F_i = F_{i-1} + F_{i-2}, \quad F_1 = F_2 = 1$$

In 1960 Donald Wall of IBM discovered that $F_n \; mod \; m$ repeats:

n	1	2	3	4	5	6	7	8	9	10	11	12	13
F_n	1	1	2	3	5	8	13	21	34	55	89	144	233
$F_n \; mod \; 11$	1	1	2	3	5	8	2	10	1	0	1	1	2

Let $K(m)$ the length of the repeated subsequence. As we can see, $K(11) = 10$.

Now consider the non-Fibonacci sequence:

$$G_i = G_{i-1} + G_{i-2}, \quad 0 \leq G_1, G_2 \leq 100,$$

Given integer values for G_1, G_2, and m determine if the sequence of $G_i \; mod \; m$ values repeats, and with what sequence length, $K(m)$, $(1 \leq G1, G2 \leq 100)$, $(2 \leq m \leq 20)$

Input:
The first line of input contains a single decimal integer $P, (1 \leq P \leq 500)$, which is the number of datasets that follow. Each dataset should be processed identically and independently. Each dataset consists one line of input containing the problem number, starting at 1, then follows the space-separated integer values of G_1, G_2 and m.

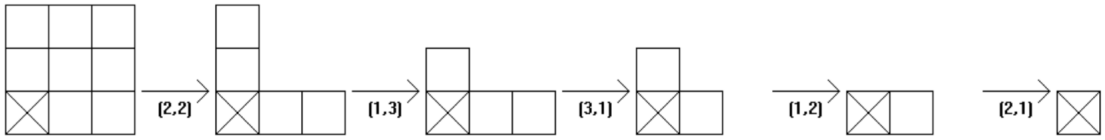
Output:
For each dataset there will be one line of output. That line will begin with the problem number, then the value of $K(m)$. If $K(m) > 100$ output 100 as its value.

Sample Input	Output for Sample Input
3	1 10
1 1 1 11	2 20
2 2 4 10	3 16
3 1 3 7	

H Chomp

Data File: H.txt
Runtime allowed = 10 seconds

Chomp is a two-player game played using a rectangular chocolate bar made up of smaller squares (cells). The players take turns choosing one cell and eating it together with those cells that are above it and to its right. (this is a low sugar chocolate bar). The bottom left cell is *poisoned* and whoever is forced to eat it loses. The following diagram shows a game during a sequence of moves:



A position in a 3-by-100 CHOMP game is described by the number p of squares in the bottom row, the number q of squares in the middle row, and the number r of squares in the top row with:

$$100 \geq p \geq q \geq r \geq 0$$

Write a program that determines whether a given position is a winning or losing position. If it is a winning position, give the next square to eat.

Input:
The first line of input contains a single decimal integer $P, (1 \leq P \leq 500)$, which is the number of datasets that follow. Each dataset should be processed identically and independently. Each dataset consists of a single line of input containing the dataset number K followed by the counts $100 \geq p \geq q \geq r \geq 0$ of squares in the bottom row (p), middle row (q), and top row (r). respectively, separated by single spaces.

Output:
For each dataset there is a single line of output. If the input position is a losing position, the output line consists of the dataset number K, followed by a single space, followed by the capital letter 'L'. Otherwise the output line consists of the dataset number K, followed by the capital letter 'W', followed by the column numbder and row number of the square to eat which results in a losing position for opponent.

Sample Input	Output for Sample Input
4	1 W 2 2
1 3 3 3	2 W 3 1
2 3 1 0	3 L
3 3 2 0	4 W 51 1
4 97 64 35	

I Counting Cows Again and Again

Data File: I.txt
Runtime allowed = 10 seconds

When farmer Jack counts the cows each day, for each group of five cows, he draws four vertical lines on his notepad, and then crosses the four with a diagonal line, as illustrated below:

We can represent one of his daily entries as a string as follows: IIII/ IIII/ IIII/ III, using capital 'i' characters and forward slash characters with exactly one space character between the groups.

The total in the above examples is 18 cows.

Given such a String, can we determine if it follows the counting rules described above?

Input:
The first line of input contains a single decimal integer $P, (1 \leq P \leq 500)$, which is the number of datasets that follow. Each dataset should be processed identically and independently.

Each dataset consists of a single line of text containing the problem number, K, beginning with 1, and then the string from Joe's notebook. A correctly formatted string contains some number of *groups of five*, separated by single spaces, and then between one and four 'I' characters for the final group.

A *group of five* begins with exactly four 'I' (capital i) characters and is followed by a '/' character.

Joe has no more than 500 head of cattle.

Output:
For each dataset, output a single line of text that begins with the problem number and then one of two strings. If Joes's data is correctly formatted output "OK" followed by a space and then the total number of cows. Otherwise, output the string "Faulty format".

Sample Input	Output for Sample Input
6	1 OK 18
1 IIII/ IIII/ IIII/ III	2 Faulty format
2 IIII/ IIII/ III/ IIII/	3 Faulty format
3 /IIII /IIII III	4 Faulty format
4 IIII/ IIIII/ IIII/	5 Faulty format
5 IIII/IIII/ IIII/ III	6 Faulty format
6 iiii/ iiii/ iii	

J Freddy's Maze Quest

Data File: J.txt
Runtime allowed = 10 seconds

Freddy the frog is about to enter a contest for solving a mazes that are especially designed for frogs. He will have to solve of up to 30 mazes, but some have no solution! He needs a genius who can write a program to help him.

A frog-maze is made up of an $n \times n$ rectangular grid. The cells of the grid are populated with non-negative integers, one per square. The goal is to jump along a legitimate path from the upper left corner to the lower right corner of the maze using the fewest jumps. The integer in a square dictates the distance the frog must jump away from that square. No jump is allowed that takes Freddy over the external wall of the maze.

All jumps must be either to the right or vertically downward. Note that a cell containing a 0 is a dead end which prevents any further progress.

Consider the 4 × 4 maze shown below:

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

Possible jump sequences are E2, S3, E1; S2, S1, E3; S2, E1, E2, S1, where E means east and S means south. The smallest number of jumps is 3.

Input:
The input contains data for one to thirty mazes, followed by a final line containing only the integer '-1'. The data for a maze starts with a line containing a single positive integer n, $4 \leq n \leq 34$, which is the number of rows in this maze. This is followed by n rows of data. Each row contains n single digits in ['0'...'9'], with no spaces between them.

Output:
The output consists of one line for each maze, containing the problem number, starting at 1, then a space, and then the smallest number of jumps to get from the upper left corner to the lower right corner. If no path is possible, output the phrase "No Path".

Sample Input	Output for Sample Input
4 2331 1213 1231 3110 4 3332 1213 1232 2120 5 11111 11111 11111 11111 11111 -1	1 3 2 No Path 3 8