Problem Set

Think.

Create.

Solve.

2016 Asia-Manila Regional
Programming Contest
December 14–16, 2016
Ateneo de Manila University

Contents

Notes

- Many problems have large input file sizes, so we suggest using fast I/O methods. For example:
  - In Java, take your input with `BufferedReader` and buffer your output with `StringBuilder`.
  - In C/C++, take your input with `scanf` and write your output with `printf`.
- Practice writing the fast I/O methods above now, so that you can save some (important) minutes in the real contest tomorrow!
- The problems are solvable in C, C++ and Java, but there's no guarantee that there's a solution in Python that passes the time limit.
- Please feel free to ask clarifications for any verdict that you feel was incorrectly given.
- The problems for the practice round were all borrowed from NOI.PH. See https://noi.ph/past-problems/ for the complete list. They are all hosted in HackerRank and everyone and their friends are invited to try them out for practice!
- Good luck!

Practice Problem A
A Simple Problem.
Time Limit: 3 seconds

Given a positive integer N, what is the minimum positive integer K such that K is a multiple of the square of N?

Note that a is a multiple of b if $a = b \cdot k$ for some integer k.

Input

The first line of input will contain a single integer T, the number of test cases.

Each test case consists of a single line, containing the integer N.

Output

For each test case, output a single line containing the integer K, the answer for that test case.

Constraints

$1 \leq T \leq 200000$
$1 \leq N \leq 200000$

Sample Input	Sample Output
1 1	1

This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`)

Practice Problem B
A Simple Problem!
Time Limit: 3 seconds

Given a positive integer N, what is the minimum positive integer K such that $K!$ is a multiple of the square of $N!$?

Note that a is a multiple of b if $a = b \cdot k$ for some integer k.

Moreover, note that for any positive integer M, $M!$ is the product of all positive integers whose value is at most M.

Input

The first line of input contains T, the number of test cases. The following lines describe the test cases.

Each test case consists of one line containing a single integer, N.

Output

For each test case, print a single integer which is the answer for that test case.

Constraints

$1 \leq T \leq 200000$
$1 \leq N \leq 200000$

Sample Input	Sample Output
5 4 5 7 11 24	8 10 14 22 48

This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`)

Practice Problem C
Tataramon
Time Limit: 3 seconds

Two of your friends, Jessie and James, are hunting *Tataramon*. Tataramon are little creatures that live around the village. To capture them, one must fight them and try to put them in a Tataraball.

Prepare for Tataraball! Before going around the village to look for the Tataramon, the computer in their rocket was able to predict which Tataramons they will encounter in their path. Note that each type of Tataramon has a unique number, called its ID. For example, #025 is called Pandan-kachu.

Make It Do-Double! Jessie and James just plan to take at most two of each kind of Tataramon. After all, they don't want to be too greedy. Perhaps some boy with a red cap will want to look for Tataramon after they do. Moreover, at the end of the day, they would want the sum of all the IDs of the Tataramon they have captured to be as large as possible.

Which Tataramons should they get and in what order will they be able to obtain them? Start coding and let your programs run at the speed of light, in order to figure out the Tataramons which they should fight.

Input

The first line of input contains T, the number of test cases.

The first line of each test case contains a single integer, N. The second line of each test case contains N integers $A_1, A_2, \ldots, A_N$, separated by single spaces, representing the IDs of the Tataramons in the order in which Jessie and James will encounter them.

Output

For each test case, output a single line consisting of space-separated integers $B_1, B_2, \ldots, B_M$, the ID of the Tataramons which Jessie and James should capture, based on the system described in the problem statement. Moreover, they should appear in the same order as they appeared in the input.

In other words, $B_1, \ldots, B_M$ must be a *subsequence* of the original sequence $A_1, \ldots, A_N$.

Take note that this subsequence is not necessarily unique. See sample output and explanation for more details.

Constraints

$1 \leq T \leq 10$
$1 \leq A_i \leq 10^9$
$1 \leq N \leq 100000$

Sample Input

```
2
4
60 52 60 60
3
9 17 1
```

Sample Output

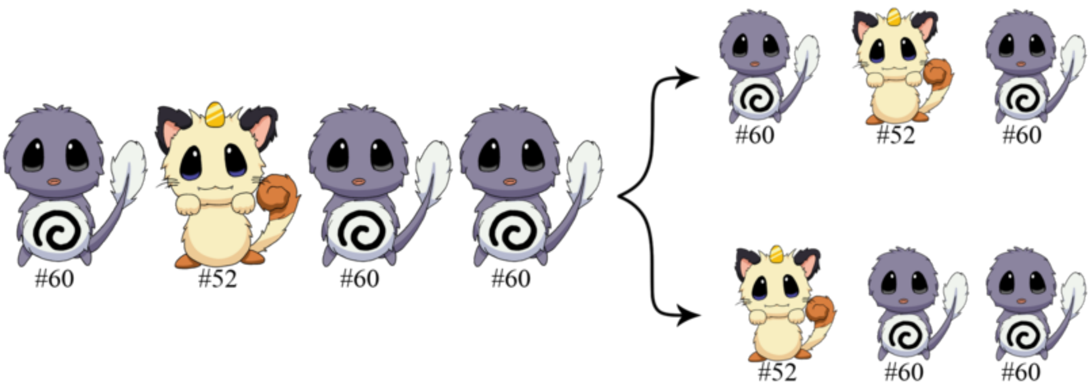
```
60 52 60
9 17 1
or
52 60 60
9 17 1
```

Explanation

For the first test case, according to the sample input, the first Tataramon they encounter will be Po-Libon (#60). They can either choose to capture this Po-Libon or move on, as they will still be able to capture enough Tataramons later to obtain the maximum possible sum.

The first sample output represents the case where they choose to immediately capture the first Po-Libon and the second sample output represents the case where they choose to not capture the first Po-Libon (and capture the next three Tataramon they encounter). Both of these output will be accepted since they will end up with a sum of $60+60+52 = 172$.

The following image illustrates the first test case:



For the second test case, Jessie and James will encounter Buhi-lastoise (#09), Partido (#17) and Naga-saur (#01), in that order. Hence, they will capture them in that order. And this is what will let them end up with the maximum sum $9+17+1 = 27$ in this test case.

This problem was borrowed from NOI.PH (https://noi.ph/past-problems/)

Practice Problem D
Friday
Time Limit: 3 seconds

- December 21, 2012 - The predicted date for the end of the world is a Friday. This is when judgment day starts and a lot of things happen every Friday since this date.
- December 28, 2012 - Putin signs a law that bans Americans from adopting Russian children.
- January 4, 2013 - This day was filled with love. Hearts around the world were truly singing.
- January 11, 2013 - The world discovers confirmation bias.
- January 18, 2013 - An indecisive girl finally chooses to sit in the backseat. Aside from that, things were quite fun fun fun.
- January 25, 2013 - People around the world did not hesitate on this day, considering the day before didn't even start.

The list goes on! Weird things happen on Fridays after the 21st of December! You want to be prepared. You should make a schedule! You list down all Fridays (in chronological order) after the 21st of December. These are the days where you intend to wear a tinfoil hat to prevent things from happening to you.

Since you are very paranoid, you need to keep track if your list is correct. You don't want to be unprotected during the dreaded Fridays of the future. You recall that leap years are those years divisible by 4, except those divisible by 100, but including those divisible by 400. You also assume that the universe will not end in 10^{18} days, and this rule about leap years will still be in place by then!

You also recall the order of the months. You can't be too careful! Here is a list of the months (in order) and the number of days it has per year.

- January (31)
- February (28. 29 during leap years.)
- March (31)
- April (30)
- May (31)
- June (30)
- July (31)
- August (31)
- September (30)
- October (31)

- November (30)
- December (31)

And since you really want to make sure, you decide to make a program to check if you listed everything correctly!

Input

The first line of input contains T, the number of test cases. The following lines describe the test cases.

Each test case consists of a single line containing a single integer, N.

Output

For each test case, print which date must be the N^{th} entry in the list.

The format must be "`MMM DD, YYYY`" (without quotes) where `MMM` is the name of the month (full month name and correctly capitalized, as in the above list and in the sample output), `DD` is the date and `YYYY` is the year (this may contain more than four digits).

Constraints

$1 \leq T \leq 50000$
$1 \leq N \leq 10^{15}$

Sample Input	Sample Output
5 1 2 3 5 1234567890123	December 28, 2012 January 4, 2013 January 11, 2013 January 25, 2013 October 8, 23660926551

This problem was borrowed from NOI.PH (https://noi.ph/past-problems/)

Practice Problem E
A Packing Problem
Time Limit: 2 seconds

During his childhood, Miguel always found inspiration as he watched his mother pack her sheets. As he grew older, Miguel took a job which is quite similar — packing cans.

Miguel's job involves packing cans in a box with a rectangular base. Miguel wonders if he can pack together two cans with different radiuses in a box. We can assume that the cans are right circular cylinders of differing heights. Miguel feels he's *packing nice* if both cans are standing upright (i.e. you should see the circular bases if you look at the box from the top) and he feels he's *packing well* if no can is on top of the other. It is okay for Miguel to pack cans such that they touch each other on the sides. The cans can even touch the sides of the box. In fact, having them touch is usually a good idea since it means that the packing is more compact. Moreover, you may assume that the height of the box is greater than the height of any of the two cans to be packed. Miguel feels he's *packing good* whenever he simultaneously feels he's *packing nice* and *packing well*. Otherwise, he feels he's *packing bad*.

Given the radiuses of their bases, can he arrange the cans in the box such that he feels he's packing good?

Input

The first line of input contains T, the number of test cases.

The next T lines each contain four positive integers L, W, R_1, R_2, separated by single spaces. L and W are the dimensions of the base of the box, and R_1 and R_2 are the radiuses of the two cans to be packed.

Output

For each test case, output a single line containing `PACKING GOOD` if Miguel feels he's packing good. Otherwise, output a single line containing `PACKING BAD`.

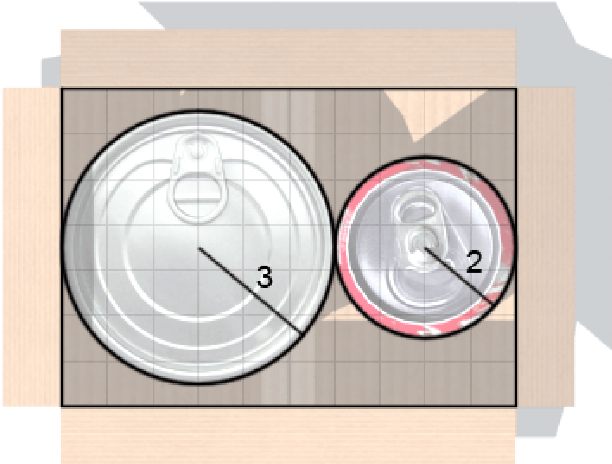
Constraints

$1 \leq T \leq 10^5$
$1 \leq L \leq 10^4$
$1 \leq W \leq 10^4$
$1 \leq R_1 \leq 10^4$
$1 \leq R_2 \leq 10^4$

Sample Input	Sample Output
2 7 10 3 2 4 7 2 2	PACKING GOOD PACKING BAD

Explanation

For the first test case, the packing can be done as follows:

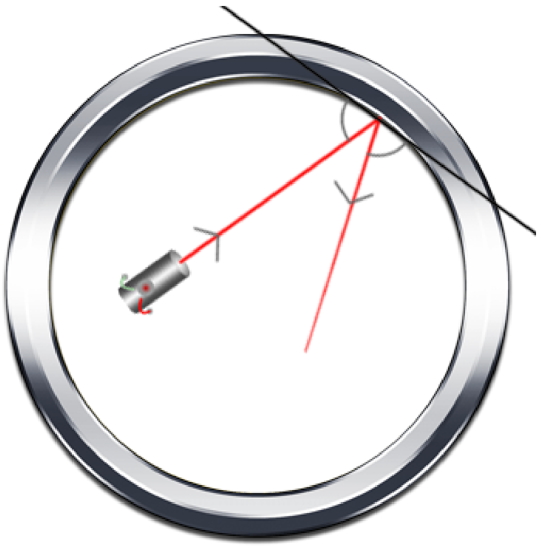


This problem was borrowed from NOI.PH (https://noi.ph/past-problems/)

Practice Problem F
Bounce Bounce Bounce
Time Limit: 3 seconds

Fidel invented a circular ring. Its boundary is made of a reflective material and at one point on the circle's boundary, a laser is placed. He wishes to set the angle of the laser in such a way that it bounces N times around the circle then returns to its original position. How many ways can he choose the initial direction of the laser?

Assume that the angle of incidence is equal to the angle of reflection (see the image below; the location of the laser in the image is for demonstration purposes only).



Input

The first line of input contains T, the number of test cases.

A test case consists of a single integer, N, on a line by itself.

Output

For each test case, output one line containing the number of ways to launch the laser so that it bounces exactly N times and returns at the exact same point.

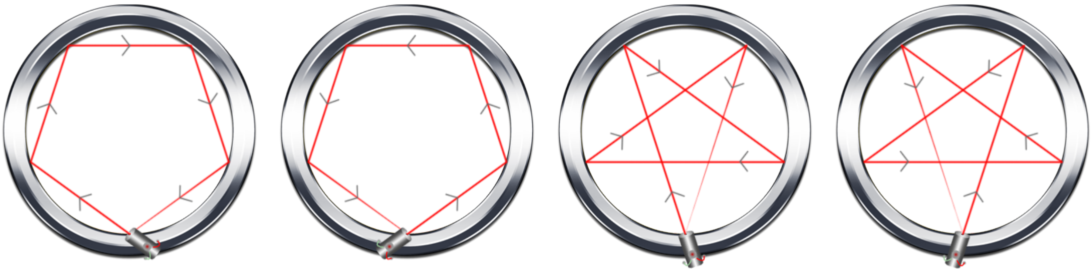
Constraints

$1 \leq T \leq 3000$
$1 \leq N \leq 10^9$

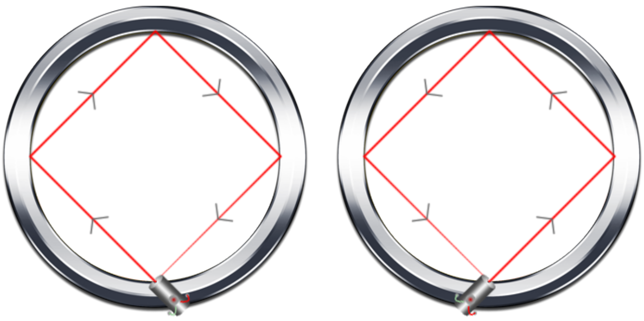
Sample Input	Sample Output
2 4 3	4 2

Explanation

For the first example, there are exactly four ways to bounce exactly four times and return to the same point. See the figure below.



For the second example, there are exactly two ways to bounce exactly three times and return to the same point. See the figure below.



This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`)

Practice Problem G
Paasa Numbers
Time Limit: 3 seconds

Pafall numbers are positive integers for which every digit is strictly less than the digit on its immediate left (if such a digit exists). These numbers get their namesake from the fact that the values of the digit seem to 'fall' as you read them from left to right. For example, 987654321 is a pafall number.

Paasa numbers are positive integers closely related to *Pafall numbers*. However, they sometimes go 'steady' when you read them from left to right. Formally speaking, a paasa number is an integer for which every digit is less than or equal to the digit on its immediate left (if such a digit exists). For example, Carly's number, 9987765321, is a paasa number.

As you would have discovered from experience by now, a number may be *paasa* but not *pafall*. But it is sure that when it is *pafall*, then it's certainly *paasa*.

Paasa numbers are generally to be avoided. That's why you decide to make a list of all the *paasa* numbers in the world so that people would be aware which ones to avoid. And you list them in increasing order. This will be a very nice list to have, especially considering the date today. We wouldn't ask you for the whole list because in doing so, you will *need some space*. We will just ask you what the N^{th} *paasa number* is.

Input

The first line of input contains T, the number of test cases. The following lines describe the test cases.

Each test case consists of one line containing a single integer, N.

Output

For each test case, print a single integer, the N^{th} paasa number.

Constraints

$1 \leq T \leq 10^4$
$1 \leq N \leq 10^{18}$

Sample Input	Sample Output
5 11 12 13 14 15	11 20 21 22 30

This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`)

Practice Problem H
Guardians of the Lunatics
Time Limit: 7 seconds

You are in charge of assigning guards to a prison where the craziest criminals are sent. The L cells form a single row and are numbered from 1 to L. Cell i houses exactly one lunatic whose *craziness level* is C_i.

Each lunatic should have one guard watching over him/her. Ideally, you should have one guard watching over each lunatic. However, due to budget constraints, you only have G guards to assign. You have to assign which lunatics each guard should watch over in order to minimize the total risk of having someone escape.

Of course, you should assign each guard to a set of adjacent cells. The *risk level* R_i that the lunatic in cell i can escape is given by the product of his/her craziness level C_i and the number of lunatics the guard assigned to him/her is watching over. Getting the sum of the R_i's from $i = 1$ to $i = L$ will give us the total amount of risk, R, that a lunatic might escape.

Given L lunatics and G guards, what is the minimum possible value of R?

Input

The first line of input contains a single integer, T, denoting the number of test cases.

The first line of each test case contains two space-separated positive integers: L and G, the number of lunatics and the number of guards respectively. The next L lines describe the craziness level of each of the lunatics. The i^{th} of these L lines describe the craziness level of the lunatic in cell block i.

Output

For each test case, output a line containing the minimum possible value of R.

Constraints

$1 \leq T \leq 22$
$1 \leq L \leq 8000$
$1 \leq G \leq 800$
$1 \leq C_i \leq 10^9$

Sample Input	Sample Output
1 6 3 11 11 11 24 26 100	299

This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`)

Practice Problem I
The Grand Noi and ICPC Battle
Time Limit: 2 seconds

The Nois are an ancient and cultured race who have a strange fascination in mathematics and computers. They are a peaceful race, but they are now engaged in an epic war against their feared foes, the Inter-continental Prairie Corgis (known as the ICPCs in short). The war has been in progress for years, and neither side has managed to make footholds in their opponent's territory.

But now the Nois have a grand plan to finally end the war and return to their normal activities of discovering new theorems and inventing math problems. The Nois, with their devastating intellects, have managed to create a superweapon that will blast the ICPCs into an alternate dimension where math and computers do not exist (the very thought that such a dimension exists chills their bones, and the poor Noi who discovered the dimension is now locked in a psychiatric ward; but that's for another story). But alas, ICPC spies discovered their plan, and a saboteur managed to sneak in and ruin the configuration of the Noi superweapon.

The superweapon is configured using a sequence of N integers, and the Noi have devised a way to rapidly test for the right configuration. However, the only Noi who can write code to perform the tests is in a psychiatric ward, and they now desperately need help. The Noi have heard of your programming prowess, and have enlisted your help in reconfiguring their superweapon.

They sent you this message:

Help us, oh great one! We need a program to help defeat the dastardly ICPCs. When looking for the right configuration of our superweapon, we perform either of two operations on our configuration sequence (which we will refer to as A): we either set $A_L, A_{L+1}, \ldots, A_{R-1}, A_R$ to V (for some integers L and R), or we try to find the sum of all $A_i \cdot A_j \cdot A_k$ for all indices i, j, k such that $L \le i < j < k \le R$ (for some integers L and R).

Input

The first line of input contains two integers, N and Q, separated by a single space, where N is the length of the configuration sequence, and Q is the number of operations that the Noi have to perform.

The next Q lines contain the operations to be performed on the configuration sequence in order, and are in either of the two following formats:

- `SET L R V` - Set all A_L, A_{L+1}, ..., A_{R-1}, A_R to V.
- `ASK L R` - Output the sum of all $A_i \cdot A_j \cdot A_k$ for all indices i, j, k such that $L \le i < j < k \le R$. Output the answer *modulo* 10^8.

Every element of the configuration sequence is initially set to 1.

Output

Output M lines, where M is the number of operations of the second kind. For each such line, output a single integer S, which is the sum of all $A_i \cdot A_j \cdot A_k$ for all indices i, j, k such that $L \leq i < j < k \leq R$, modulo 10^8.

Constraints

$1 \leq N \leq 10^9$
$1 \leq Q \leq 10^5$
$1 \leq L \leq R \leq N$
$1 \leq V \leq 10^6$

Sample Input	Sample Output
10 9 ASK 1 4 ASK 3 10 SET 5 10 3 ASK 1 4 ASK 3 10 SET 4 6 10000 ASK 1 4 ASK 3 10 ASK 3 4	4 56 4 828 30001 1980162 0

This problem was borrowed from NOI.PH (`https://noi.ph/past-problems/`*)*