Greetings From Globussoft

- Given below are 5 Programming questions, you have to solve any 3 out of 5 questions.

- These 5 questions you can attempt in any technology like C/C++, java, .Net, PHP

- To solve these 3 questions you've max. 3 hours.

- While Solving these questions you are not allowed to use any **Search Engine** like Google, Yahoo, Bing …

All the best for your test

Globussoft

QUESTION - 1

One of the famous proofs of modern mathematics is Georg Cantor's demonstration that the set of rational numbers is enumerable. The proof works by using an explicit enumeration of rational numbers as shown in the diagram below.

```
1/1 1/2 1/3 1/4 1/5 ...
2/1 2/2 2/3 2/4
3/1 3/2 3/3
4/1 4/2
5/1
```

In the above diagram, the first term is 1/1, the second term is 1/2, the third term is 2/1, the fourth term is 3/1, the fifth term is 2/2, and so on.

Input

The input starts with a line containing a single integer **t** <= 20, the number of test cases. **t** test cases follow.

Then, it contains a single number per line.

Output

You are to write a program that will read a list of numbers in the range from 1 to 10^7 and will print for each number the corresponding term in Cantor's enumeration as given below.

Example

Input:

```
3
3
14
7
```

Output:

```
TERM 3 IS 2/1
TERM 14 IS 2/4
TERM 7 IS 1/4
```

QUESTION – 2

Yesterday was Sam's birthday. The most interesting gift was definitely the chessboard. Sam quickly learned the rules of chess and defeated his father, all his friends, his little sister, and now no one wants to play with him any more.

So he decided to play with another birthday gift – a Book of Math Problems for Young

Mathematicians. He opened the book somewhere in the middle and read the following problem: "How many knights can be placed on a chessboard without threatening each other?" After a while he realized that this was trivial and moved on to the next problem: "How many bishops can be placed on a chessboard without threatening each other?". Sam is in trouble here. He is not able to solve this problem and needs your help.

Sam's chessboard has size N *x* N. A bishop can move to any distance in any of the four diagonal directions. A bishop threatens another bishop if it can move to the other bishop's position. Your task is to compute the maximum number of bishops that can be placed on a chessboard in such a way that no two bishops threaten each other.

Input

The input file consists of several lines. The line number i contains a single number N representing the size of the i-th chessboard. [N <= 10^100]

Output

The output file should contain the same number of lines as the input file. The i-th line should contain one number – the maximum number of bishops that can be placed on i-th chessboard without threatening each other.

Example

Input:
```
2
3
```

Output:
```
2
4
```

QUESTION – 3

In Byteland they have a very strange monetary system.

Each Bytelandian gold coin has an integer number written on it. A coin n can be exchanged in a bank into three coins: n/2, n/3 and n/4. But these numbers are all rounded down (the banks have to make a profit).

You can also sell Bytelandian coins for American dollars. The exchange rate is 1:1. But you can not buy Bytelandian coins.

You have one gold coin. What is the maximum amount of American dollars you can get for it?

Input

The input will contain several test cases (not more than 10). Each testcase is a single line with a number n, 0 <= n <= 1 000 000 000. It is the number written on your coin.

Output

For each test case output a single line, containing the maximum amount of American dollars you can make.

Example

Input:
```
12
2
```

Output:
```
13
2
```

QUESTION – 4

Farmer John is making the difficult transition from raising mountain goats to raising cows. His farm, while ideal for mountain goats, is far too mountainous for cattle and thus needs to be flattened out a bit. Since flattening is an expensive operation, he wants to remove the smallest amount of earth possible.

The farm is long and narrow and is described in a sort of two-dimensional profile by a single array of N (1 <= N <= 1000) integer elevations (range 1..1,000,000) like this:

1 2 3 3 3 2 1 3 2 2 1 2,

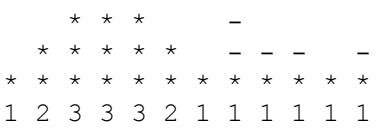
which represents the farm's elevations in profile, depicted below with asterisks indicating the heights:

```
    * * *     *
  * * * * *   * * *   *
* * * * * * * * * * * *
1 2 3 3 3 2 1 3 2 2 1 2
```

A contiguous range of one or more equal elevations in this array is a "peak" if both the left and right hand sides of the range are either the boundary of the array or an element that is lower in elevation than the peak. The example above has three peaks.

Determine the minimum volume of earth (each unit elevation reduction counts as one unit of volume) that must be removed so that the resulting landscape has no more than K (1 <= K <= 25) peaks. Note well that elevations can be reduced but can never be increased.

If the example above is to be reduced to 1 peak, the optimal solution is to remove 2 + 1 + 1 + 1 = 5 units of earth to obtain this set of elevations:



where '-'s indicate removed earth.

Input

The first line of the input contains integer t representing the number of test cases. Then t test cases follow. Each test case has the following form:

- Line 1: Two space-separated integers: N and K
- Lines 2..N+1: Each line contains a single integer elevation. Line i+1 contains the elevation for index i.

Output

For each test case, output the minimum volume of earth that must be removed to reduce the number of peaks to K.

Example

Input:

```
1
12 1
1
2
3
3
3
2
1
3
2
2
1
2
```

Output:

```
5
```

QUESTION – 5

Fred works as an IT consultant in an insurance company. As they always had a large amount of customers waiting and arguing at the front desk, management decided to deploy a ticket machine. Each customer would get a ticket with a number and there will be fancy LCD display over each desk showing the number of the next person. Fred was appointed to get this new enhancement working.

Because Fred is lazy when it comes to manual labor and as an IT consultant he wouldn't lower himself to the level of some hardware technician (except when upgrading his own computer), he asked few technicians to install the displays and prepared himself just to plug in the ticket machine and try it out. Unfortunately (for Fred) the technicians, either inspired by Mr.Bean or because of their carelessness, installed the display upside-down.

Being a software guy, Fred decided that the hardware should not be tampered with after it is installed (except for the case if he would be able to get back the technicians to repair it, but they were already angry at him for his nagging). Then he noted that from time to time the display shows a correct number even when it is upside-down. And hey, the ticket machine is an embedded device and contains a small processor! It would be just a sin for an IT guy not to try to meddle with it and try running an own version of Linux. Now we just need to figure out which readable numbers will the display show.

Task specification

In the beginning the display shows the number 1 on its display. Each second the number shown is increased by 1. We see the display upside-down and thus not everything we see will make sense. Your task is to compute the K-th valid number we will see on the display. The digits the display uses are shown on the images below. An upside-down 1 still count as 1. The number we see may have leading zeroes – e.g. turning the number 600 upside down leads to a valid number.

Input

t - the number of test cases [t <= 2200], than t test cses follows. Each test case consist of one integer Ki [0 < Ki <= 10^200].

Output

For each Ki from the input file, output the Ki-th number shown on the display (including the leading zeroes, if there are some).

Example

Input:

```
8
1
2
3
4
5
6
8
98
```

Output:

```
1
2
5
9
8
6
11
002
```