100!

Description

Given a number n, please calculate the number of trailing 0s of $n!$

For example, given $n = 10$, we have $10! = 3628800$. Thus, we should output 2 since there are 2 trailing zeros in 3628800.

Input

Given an integer T indicates the number of testcases.

For the next T lines, there is a number n $(0 <= n <= 20000)$ per line.

Output

Output the number of trailing 0s of $n!$ for each testcases.

Sample Input 1

```
5
10
13
15
16
18
```

Sample Output 1

```
2
2
3
3
3
```

Hint

老師上課的題目^^

想想幾乘幾的時候數字後面會有0出現?

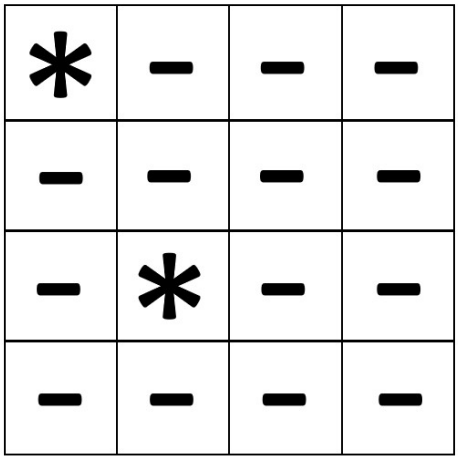
Best destination for stargazing

Description

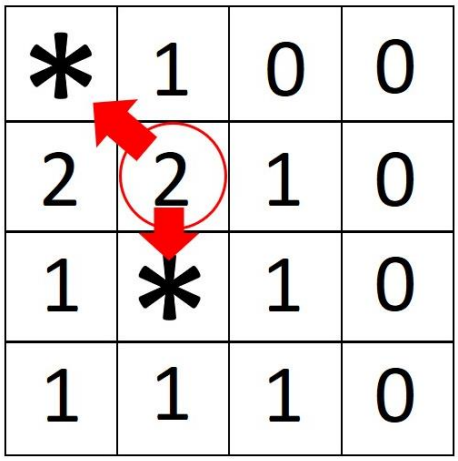
When was the last time you saw more than five stars in the night sky? The night sky is like a giant puzzle, hidden among the thousands of stars you can find dozens of constellations. Stars don't disappear; th ey keep blazing, even when the night is over. However, a good place to go stargazing is importance. As our population grows, so too does light pollution, clouding the atmosphere and all that lies beyond it.

Today, as few as 500 stars are visible from many urbanareas. Your goal is to find an area to see if the place is suitable for going stargazing.

Suppose the following 4 × 4 field with 2 starts (which are represented by an '*' character):



If we would represent the same field placing the hint numbers described above, we would end up with:



As you may have already noticed, each square may have at most 8 adjacent squares.

Input

The input will consist of an arbitrary number of fields. The first line of each field contains two integers n and m (0< n, m ≤ 100) which stands for the number of lines and columns of the field respectively.

The next n lines contain exactly m characters and represent the field. Each safe square is represented by an '-' character (without the quotes) and each star square is represented by an '*' character (also with out the quotes).

The first field line where n = m = 0 represents the end of input and should not be processed.

Output

For each field, you must print the following message in a line alone:

Field #x:

Where x stands for the number of the field (starting from 1).

The next n lines should contain the field with the '-' characters replaced by the number of adjacent starts to that square. There must be an empty line between field outputs.

Sample Input 1

```
4 4
*---
----
-*--
----
3 5
**---
-----
-*---
0 0
```

Sample Output 1

```
Field #1:
*100
2210
1*10
1110

Field #2:
**100
33200
1*100
```

Bangla Numbers

Description

Bangla numbers use "kuti" (10000000), "lakh" (100000), "hajar" (1000), "shata" (100) while expanding and converting to text.You are going to write a program to convert a given number to text with them.

Input

The input file may contain several test cases. Each case contains a non-negative number n. $(0 \leq n \leq 999999999999999)$.

The line where $n = -1$ represents the end of input and should not be processed.

Output

For each case of input, you must output a line starting with the case number followed by the converted text. Each number and word are followed by one blank space but not included the last one.

Sample Input 1

```
23764
45897458973958
200002
-1
```

Sample Output 1

```
1. 23 hajar 7 shata 64
2. 45 lakh 89 hajar 7 shata 45 kuti 89 lakh 73 hajar 9 shata 58
3. 2 lakh 2
```

Hint

以中文百千萬單位為例:

"萬" (10000), "千"(1000), "百" (100)

12 => 12

123 => 1百23

1234 => 1千2百34

12345678 => 1千2百34萬5千6百78

Do Not Access

Description

DNA (Deoxyribonucleic Acid) is the molecule which contains the genetic instructions. It consists of four different nucleotides, namely Adenine, Thymine, Guanine, and Cytosine as shown in Figure1. If we repr esent a nucleotide by its initial character, a DNA strand can be regarded as a long string (sequence of characters) consisting of the four characters A, T, G, and C. For example, assume we are given some pa rt of a DNA strand which is composed of the following sequence of nucleotides:

"Thymine-Adenine-Adenine-Cytosine-Thymine-Guanine-CytosineCytosine-Guanine-Adenine-Thymine"

Then we can represent the above DNA strand with the string "TAACTGCCGAT."

The biologist Prof. Ahn found that a gene Xcommonly exists in the DNA strands of five different kinds of animals, namely dogs, cats, horses, cows, and monkeys. He also discovered that the DNA sequences of the gene X from each animal were very alike. See Figure 2.

Prof. Ahn thought that humans might also have the gene X and decided to search for the DNA sequence of X in human DNA. However, before searching, he should define a representative DNA sequence of gene X because its sequences are not exactly the same in the DNA of the five animals. He decided to use the Hamming distance to define the representative sequence.

The Hamming distance is the number of different characters at each position from two strings of equal length. For example, assume we are given the two strings "AGCAT" and "GGAAT."The Hamming distanc e of these two strings is 2because the 1st and the 3rd characters of the two strings are different. Using the Hamming distance, we can define a representative string for a set of multiple strings of equal length. Given a set of strings $S = s1, ..., s_m$ of length n, the consensus error between a string y of length n and the set Sis the sum of the Hamming distances between y and each si in S.

If the consensus error between y and S is the minimum among all possible strings y of length n, y is called a consensus string of S. For example, given the three strings "AGCAT" "AGACT" and "GGAAT" the consensus string of the given string sis "AGAAT" because the sum of the Hamming distances between "AGAAT" and the three strings is 3 which is minimal. (In this case, the consensus string is unique, but in general, there can be more than one consensus string.) We use the consensus string as a representative of the DNA sequence. For the example of Figure 2 above, a consensus string of gene X is "GCAAAT GGCTGTGCA" and the consensus error is 7.

Input

Your program is to read from standard input. The input consists of T test cases. The number of test cases T is given in the first line of the input. Each test case starts with a line containing two integers m and n which are separated by a single space. The integer m (4 ≤m ≤ 50) represents the number of DNA sequences and n (4 ≤ n ≤ 1000) represents the length of the DNA sequences, respectively. In each of the n ext m lines, each DNA sequence is given.

Output

Your program is to write to standard output. Print the consensus string in the first line of each case and the consensus error in the second line of each case. If there exists more than one consensus string, pri nt the lexicographically smallest consensus string.

Notice that there is a blank line at the end of output.

Sample Input 1

```
3
5 8
TATGATAC
TAAGCTAC
AAAGATCC
TGAGATAC
TAAGATGT
4 10
ACGTACGTAC
CCGTACGTAG
GCGTACGTAT
TCGTACGTAA
6 10
ATGTTACCAT
AAGTTACGAT
AACAAAGCAA
AAGTTACCTT
AAGTTACCAA
TACTTACCAA
```

Sample Output 1

```
TAAGATAC
7
ACGTACGTAA
6
AAGTTACCAA
12
```

Hint

Human is similar to monkeys, since both of them are not able to completely read the description.

Chihuahua Loves to Eat

Description

A Cute Chihuahua was trapped by the evil witch in the magic tower because of his want to eat some pet food. The rooms here have some very special rules:

> 1. Each room has its own number and leads to another room with specific room number
> 2. Every room has a pack of pet food to collect
> 3. If the room number n is negative, it leads to the room number `n*n`
> 4. If the room number n is a positive even number, it leads to the room number `n/2`
> 5. If the room number n is a positive odd number, it leads to the room number `3n+1`
> 6. There is no room number 0

The only way to escape from the magic tower is:

Go to room number 1

Chihuahua sees a continuous range of room entry numbers (L<=n<=U) now. It wants to get out of this tower as soon as possible

However, It is greedy and wants to get the most pet food!

Please help it to choose the room entrance that will get the most pet food!

Input

Each line gives you two numbers L, U, which are the lower and upper bounds for room entrance numbers.

L stands for lower bound (inclusive)
U stands for upper bound (inclusive)

The two numbers will be between -100 and 10000.

Output

Each line gives you a string and an integer.

The string indicates the room number Chihuahua choose at first.
The next integer represents the number of packs of pet food that Chihuahua collected.

Notice that there is a blank line at the end of output.

Sample Input 1

```
1 10
100 200
201 210
900 1000
```

Sample Output 1

```
Room_9 20
Room_171 125
Room_206 89
Room_937 174
```

Teach Otto a Lesson

Description

> You may love your pet, but don't use its name in your password

We often use passwords with low security, such as birthdays, phone numbers, and names, but this also makes it easy for hackers to guess passwords and do something bad.

Otto, who has researched in information security, felt that he couldn't go on like this! However, the country is easy to change, but the nature is hard to change!

He still invents his ciphers using only numbers, and makes them fit his "Perfect pattern", which he thinks is secure enough.

He defines "Perfect pattern" as:

The even index(include 0)of the password string must be even, and the odd index must be prime

No matter how long the password is, let's teach him a lesson! Show him how insecure the password he invented is!

We define the "Hacking Factor" as:

(# of all passwords / # of target passwords) mod (1234567890)

For instance, given $length = 3$, we have $10^3 = 1000$ combinations of all passwords, and there are $5 \times 4 \times 5 = 100$ passwords which match the perfect pattern, thus the Hacking Factor is 10.

Input

Each row has a number N, which represents the length of the password that Otto wants to take.

Note:

0 <= N <= 1e15

Output

One output per row, corresponding to the "hacking factor" of the entered password length.

Note:

If the password length is unreasonable, the hacking factor is 0

There is a blank line at the end of output.

Sample Input 1

```
1
2
3
50
```

Sample Output 1

```
2
5
10
663038555
```

Hint

1. Is there a formula we can use to find the hacking factor?
2. $(x \cdot y)\ mod\ M = ((x\ mod\ M) \cdot (y\ mod\ M))\ mod\ M$