**First：月之数**

**Time Limit: 1000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 10531    Accepted Submission(s): 6175**

**Problem Description**

当寒月还在读大一的时候，他在一本武林秘籍中（据后来考证，估计是计算机基础，狂汗-ing），发现了神奇的二进制数。  
如果一个正整数m表示成二进制，它的位数为n（不包含前导0），寒月称它为一个n二进制数。所有的n二进制数中，1的总个数被称为n对应的月之数。  
例如，3二进制数总共有4个，分别是4（100）、5（101）、6（110）、7（111），他们中1的个数一共是1＋2＋2＋3=8，所以3对应的月之数就是8。

**Input**

给你一个整数T，表示输入数据的组数，接下来有T行，每行包含一个正整数 n（1<=n<=20）。

**Output**

对于每个n ，在一行内输出n对应的月之数。

**Sample Input**

3

1

2

3

**Sample Output**

1

3

8

**Second：Tiling\_easy version**

**Time Limit: 1000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 8447    Accepted Submission(s): 6507**

**Problem Description**

有一个大小是 2 x n 的网格，现在需要用2种规格的骨牌铺满，骨牌规格分别是 2 x 1 和 2 x 2，请计算一共有多少种铺设的方法。

**Input**

输入的第一行包含一个正整数T（T<=20），表示一共有 T组数据，接着是T行数据，每行包含一个正整数N（N<=30），表示网格的大小是2行N列。

**Output**

输出一共有多少种铺设的方法，每组数据的输出占一行。

**Sample Input**

3

2

8

12

**Sample Output**

3

171

2731

**Third：A+B for Input-Output Practice (VIII)**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 159428    Accepted Submission(s): 48172**

**Problem Description**

Your task is to calculate the sum of some integers.

**Input**

Input contains an integer N in the first line, and then N lines follow. Each line starts with a integer M, and then M integers follow in the same line.

**Output**

For each group of input integers you should output their sum in one line, and you must note that there is a blank line between outputs.

**Sample Input**

3

4 1 2 3 4

5 1 2 3 4 5

3 1 2 3

**Sample Output**

10

15

6

**Fourth：压缩问题**

**Time Limit: 1000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 8447    Accepted Submission(s): 6507**

**Problem Description**

你的任务是将一串字符实现简单的压缩

**Input**

输入一个 N ，表示要进行 N 次，在每一次中，都输入一串字符。

**Output**

输出压缩后的字符串

**Sample Input**

2

aabcccccaaa

welcometonnowcoderrrrr

**Sample Output**

a2bc5a3

welcomeron2owcoder5

**Fifth:Hello World!**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 32768/32768 K (Java/Others)  
Total Submission(s): 694    Accepted Submission(s): 275**

**Problem Description**

Your task is to print ... er ... "Hello World" ... in a fantastic way -- using a beautiful font.  
  
I've sent you a nice font for you to use, but I'm too busy to tell you how. Can you help yourself?

**Input**

The first line contains a single integer T (T <= 20), the number of test cases.   
Each case begins with an integer C (1 <= C <= 80) in a single line, then each of the following C lines contains five two-digit numbers in hex (letters will be in uppercase). Don't ask me what they mean, I'm too busy...

**Output**

For each test case, print the case number in the first line, then followed by a blank line.  
After that, print all T characters. Use a single blank column of spaces between two consecutive characters. Each line should have exactly 6C-1 character (again, don't ask me why).  
Don't forget to print another blank line after the output of each test case.

**Sample Input**

2

11

7F 08 08 08 7F

38 54 54 54 18

00 41 7F 40 00

00 41 7F 40 00

38 44 44 44 38

00 00 00 00 00

3F 40 38 40 3F

38 44 44 44 38

7C 08 04 04 08

00 41 7F 40 00

38 44 44 48 7F

5

14 08 3E 08 14

04 02 01 02 04

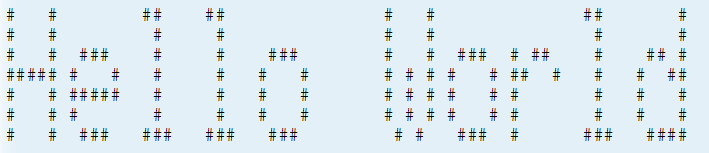
40 40 40 40 40

04 02 01 02 04

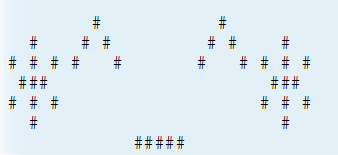
14 08 3E 08 14

**Sample Output**

Case 1:



Case 2:



**Sixth:A hard puzzle**

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 44078    Accepted Submission(s): 16041**

**Problem Description**

lcy gives a hard puzzle to feng5166,lwg,JGShining and Ignatius: gave a and b,how to know the a^b.everybody objects to this BT problem,so lcy makes the problem easier than begin.  
this puzzle describes that: gave a and b,how to know the a^b's the last digit number.But everybody is too lazy to slove this problem,so they remit to you who is wise.

**Input**

There are mutiple test cases. Each test cases consists of two numbers a and b(0<a,b<=2^30)

**Output**

For each test case, you should output the a^b's last digit number.

**Sample Input**

7 66

8 800

**Sample Output**

9

6

# Extra

# Big Number

**Time Limit: 2000/1000 MS (Java/Others)    Memory Limit: 65536/32768 K (Java/Others)  
Total Submission(s): 38050    Accepted Submission(s): 18367**

**Problem Description**

In many applications very large integers numbers are required. Some of these applications are using keys for secure transmission of data, encryption, etc. In this problem you are given a number, you have to determine the number of digits in the factorial of the number.

**Input**

Input consists of several lines of integer numbers. The first line contains an integer n, which is the number of cases to be tested, followed by n lines, one integer 1 ≤ n ≤ 107 on each line.

**Output**

The output contains the number of digits in the factorial of the integers appearing in the input.

**Sample Input**

2

10

20

**Sample Output**

7

19