# code jam
printf("hello, world!\n");

Qualification Round 2016

[A. Counting Sheep](#)

[B. Revenge of the Pancakes](#)

C. Coin Jam

[D. Fractiles](#)

[Contest Analysis](#)

Questions asked [1]

─ Submissions

Counting Sheep

7pt	Not attempted **26558/29356 users** correct (90%)
8pt	Not attempted **25729/26216 users** correct (98%)

Revenge of the Pancakes

10pt	Not attempted **22527/23686 users** correct (95%)
10pt	Not attempted **21383/22147 users** correct (97%)

Coin Jam

10pt	Not attempted **13361/15342 users** correct (87%)
20pt	Not attempted **6297/9111 users** correct (69%)

Fractiles

10pt	Not attempted **8250/9708 users** correct (85%)
25pt	Not attempted **2356/4955 users** correct (48%)

─ Top Scores

Lewin	100
Endagorion	100
xiaowuc1	100
xyz111	100
HellKitsune123	100
seanwentzel	100
ivan.popelyshev	100
burunduk3	100
Nicolas16	100
ctunoku	100

Practice Mode   Contest scoreboard | Sign in

Problem C. Coin Jam

This contest is open for practice. You can try every problem as many times as you like, though we won't keep track of which problems you solve. Read the Quick-Start Guide to get started.

Small input 10 points	[Solve C-small]
Large input 20 points	[Solve C-large]

Problem

A *jamcoin* is a string of **N** ≥ 2 digits with the following properties:

- Every digit is either 0 or 1.
- The first digit is 1 and the last digit is 1.
- If you interpret the string in any base between 2 and 10, inclusive, the resulting number is not prime.

Not every string of 0s and 1s is a jamcoin. For example, 101 is not a jamcoin; its interpretation in base 2 is 5, which is prime. But the string 1001 is a jamcoin: in bases 2 through 10, its interpretation is 9, 28, 65, 126, 217, 344, 513, 730, and 1001, respectively, and none of those is prime.

We hear that there may be communities that use jamcoins as a form of currency. When sending someone a jamcoin, it is polite to prove that the jamcoin is legitimate by including a nontrivial divisor of that jamcoin's interpretation in each base from 2 to 10. (A nontrivial divisor for a positive integer K is some positive integer other than 1 or K that evenly divides K.) For convenience, these divisors must be expressed in base 10.

For example, for the jamcoin 1001 mentioned above, a possible set of nontrivial divisors for the base 2 through 10 interpretations of the jamcoin would be: 3, 7, 5, 6, 31, 8, 27, 5, and 77, respectively.

Can you produce **J** different jamcoins of length **N**, along with proof that they are legitimate?

Input

The first line of the input gives the number of test cases, **T**. **T** test cases follow; each consists of one line with two integers **N** and **J**.

Output

For each test case, output **J**+1 lines. The first line must consist of only `Case #x:`, where x is the test case number (starting from 1). Each of the last **J** lines must consist of a jamcoin of length **N** followed by nine integers. The i-th of those nine integers (counting starting from 1) must be a nontrivial divisor of the jamcoin when the jamcoin is interpreted in base i+1.

All of these jamcoins must be different. You cannot submit the same jamcoin in two different lines, even if you use a different set of divisors each time.

Limits

T = 1. (There will be only one test case.)
It is guaranteed that at least **J** distinct jamcoins of length **N** exist.

Small dataset

N = 16.
J = 50.

Large dataset

N = 32.
J = 500.

Note that, unusually for a Code Jam problem, you already know the exact contents of each input file. For example, the Small dataset's input file will always be exactly these two lines:

```
1
16 50
```

So, you can consider doing some computation before actually downloading an input file and starting the clock.

Sample

Input	Output
1	Case #1:
6 3	100011 5 13 147 31 43 1121 73 77 629
	111111 21 26 105 1302 217 1032 513 13286 10101
	111001 3 88 5 1938 7 208 3 20 11

In this sample case, we have used very small values of **N** and **J** for ease of explanation. Note that this sample case would not appear in either the Small or Large datasets.

This is only one of multiple valid solutions. Other sets of jamcoins could have been used, and there are many other possible sets of nontrivial base 10 divisors. Some notes:

- 110111 could not have been included in the output because, for example, it is 337 if interpreted in base 3 (1*243 + 1*81 + 0*27 + 1*9 + 1*3 + 1*1), and 337 is prime.
- 010101 could not have been included in the output even though 10101 is a jamcoin, because jamcoins begin with 1.
- 101010 could not have been included in the output, because jamcoins end with 1.
- 110011 is another jamcoin that could have also been used in the output, but could not have been added to the end of this output, since the output must contain exactly **J** examples.
- For the first jamcoin in the sample output, the first number after 100011 could not have been either 1 or 35, because those are trivial divisors of 35 (100011 in base 2).

Powered by

Google Cloud Platform