"Anyone who considers arithmetic methods of producing random digits is, of course, in a state of sin."
John von Neumann, 1951.

Given a list of n real numbers, $(x_1, x_2, \ldots, x_n)$, the mean is defined as

$$m = \frac{1}{n}\sum_{i=1}^{n} x_i$$

The standard deviation is defined as the square root of

$$\frac{1}{n}\sum_{i=1}^{n}(x_i - m)^2$$

Given n and a random number generator seed, compute the standard deviation of the first n numbers returned by the generator.

The generator function is given below. I apologize to all those for whom C is not a native language.

```
unsigned long long seed;
long double gen()
{
    static const long double Z = ( long double )1.0 / (1LL<<32);
    seed >>= 16;
    seed &= ( 1ULL << 32 ) - 1;
    seed *= seed;
    return seed * Z;
}
```

Input

The first line of input gives the number of cases, N (at most 40). N test cases follow. Each one is a line containing an integer, n ($1 \leq n \leq 10,000,000$), and an integer, *seed* ($0 \leq seed < 2^{64}$).

Output

For each test case, output one line containing '`Case #`x`:`' followed by the standard deviation of the first n numbers returned by `gen()` after seed is initialized to the given value. Round the answer to 5 decimal places. Answers with absolute error of at most 10^{-4} will be deemed correct.

If you need a hint, read the problem again.

Sample Input

```
5
2 16777216
2 4294967296
10000000 0
2 2147483648
10000 382759482784958
```

Sample Output

```
Case #1: 0.00001
Case #2: 0.00000
Case #3: 0.00000
Case #4: 0.09375
Case #5: 1283729051.97967
```