A Black Box algorithm supposes that natural number sequence $u(1), u(2), \ldots, u(N)$ is sorted in non-descending order, $N \le M$ and for each p $(1 \le p \le N)$ an inequality $p \le u(p) \le M$ is valid.

Making tests for this algorithm we have met with the following problem. For setting a random sequence $\{u(i)\}$ a usual random data generator did not fit. As the sequence itself had been imposed certain restrictions, the method of choosing the next random element (in the interval defined by restrictions) did not give the random sequence as a whole.

We have come to a conclusion that the problem can be solved in the following way. If we arrange all possible sequences in certain order (for example, in lexicographical order) and assign each sequence its number, after choice of the random number it is possible to take the correspondent sequence for the random one. At the first glance it seems enough to make up a program generating all these sequences in such order. Alas! Even having not great values of M and N it would have taken any powerful modern computer centuries to enumerate all such sequences. It turned out it was possible to avoid generating all sequences if we managed to create required sequence according to its number immediately. But even this statement does not cover all. As the amount of sequences is quite large, the number can be a long one, composed of hundreds decimal digits, though our random data generator could give only normal numbers. We decided to produce a long random number from a real random number distributed in [0,1]. Namely, present the number in binary notation: $0.b_1b_2\ldots$, where all $b_i = 0$ or 1. Let us set a regulation to associate such real number to an integer from $[A, B]$ segment:

Formula

$$G(A, B, 0.b_1b_2\ldots b_p) = \begin{cases} A, & \text{if } p = 0 \text{ or } A = B; \\ \text{otherwise:} & \begin{cases} G(A, (A+B) \text{ div } 2, 0.b_2b_3\ldots b_p), & \text{if } b_1 = 0 \\ G((A+B) \text{ div } 2 + 1, B, 0.b_2b_3\ldots b_p), & \text{if } b_1 = 1 \end{cases} \end{cases}$$

Here we suppose, that $A \le B$, $p \ge 0$, and "div 2" is an integer division by 2.

Let M, N $(1 \le N \le M \le 200)$ and a binary real number $0.b_1b_2\ldots b_p$ $(1 \le p \le 400)$ be given. Write a program to find out the corresponding $u(1), u(2), \ldots, u(N)$ sequence, i.e. to find a sequence with $G(1, T, 0.b_1b_2\ldots b_p)$ number in lexicographical order of all possible $\{u(i)\}$ for the given M and N (T is the quantity of such sequences). Numeration begins with 1.

Keep in mind that in lexicographical order $\{l(i)\}$ proceeds $\{h(i)\}$ if after omitting equal beginnings, the first number of $\{l(i)\}$ tail is smaller than the first number or $\{h(i)\}$ tail.

Following example illustrates the list of all possible sequences for $M = 4$ and $N = 3$ in lexicographical order.

Example

```
1, 2, 3
1, 2, 4
1, 3, 3
1, 3, 4
1, 4, 4
2, 2, 3
2, 2, 4
2, 3, 3
2, 3, 4
2, 4, 4
3, 3, 3
3, 3, 4
3, 4, 4
4, 4, 4
```

(here T=14)

Input

The first line of the input is an integer K, then a blank line followed by K datasets. There is a blank line between datasets.

The first line of each dataset contains M and N. The second line contains binary real number $0.b_1b_2\ldots b_p$ (without leading, trailing and other spaces).

Output

For each dataset, write into the output data file the corresponding sequence $u(1), u(2), \ldots, u(N)$. The sequence numbers should be separated with spaces and end-of-line characters.

There should be up to 20 numbers in each line. If neccesary, the numbers will have leading blanks to occupy 3 characters. Print a blank line between datasets.

A note (it does not concern the solution of this task):

The choice of random binary vector $0.b_1b_2\ldots b_p$ does not give an absolute uniform random data generator if we use the Formula. However, taking into account the fact that $[A, B]$ interval is big we shall obtain a distribution applicable in most cases.

Sample Input

```
1

4 3
0.01101101011110010001101010001011010
```

Sample Output

```
  2  2  4
```