Problem 5: Pascal Key Breaker

Time limit: 2 seconds

Binomial Theorem is very useful tool in algebra and statistics. It is believed that this theorem was first introduced by a Persian and Chinese mathematician back in 13th century. The number of terms in the expansion is always $n + 1$ and the value of each term coefficient change with respect to its position in the expansion. For a higher value of n, the binomial expansion can be calculated as

$$(x + y)^n = \sum_{k=0}^{n} \binom{n}{k} x^{n-k} y^k$$

Where $\binom{n}{k}$ represent the combinations and can be calculated as below:

$$\binom{n}{k} = \frac{n!}{k!\,(n-k)!}$$

Recently, Mr. Wang was on an exploration trip in Bursa, an ancient city in Turkey. He found an interesting cupboard with several small cabinets arranged in a pyramid shape. Each cabin has a special lock and can be open with a preassigned key. Mr. Wang quickly realized that the pyramid shaped cupboard follows the famous Pascal's triangle arrangement, where each row follow the binomial expansion. As a computer programmer, Mr. Wang asked your help to find the key to the corresponding lock for the given values of x, y, and n . Further, as a helping and kind person Mr. Wang also give you a hint that the key to each cabin is the r-th term (1-indexed) of the binomial expansion according to the position of the cabin in the pyramid, calculated as

$$T_r = \binom{n}{r-1} x^{n-(r-1)} y^{r-1}$$

Where ***r-th*** term is 1-indexed, meaning the first term corresponds to $r = 1$.

Input

The first line contains the number of test cases. And for each test case the first line gets the integer term n, $(0 \le n \le 50)$, the power of binomial. The next line contains an integer r, $(1 \le r \le n + 1)$, the position of the term. The third line contain two integer numbers x and y, $(1 \le x, y \le 100)$, the components of the binomial expression, separated by single space.

Output

A single line prints the value of the ***r-th*** term.

Sample input & output

The following is an example of a sample input and corresponding correct outputs.

Sample input	Sample Output
2 5 3 2 3 4 2 1 2	720 8