For a string of $n$ bits $x_1, x_2, x_3, ..., x_n$, the adjacent bit count of the string ($\text{AdjBC}(x)$) is given by

$$x_1 \cdot x_2 + x_2 \cdot x_3 + x_3 \cdot x_4 + ... + x_{n-1} \cdot x_n$$

which counts the number of times a 1 bit is adjacent to another 1 bit. For example:

$$\text{AdjBC}(011101101) = 3$$
$$\text{AdjBC}(111101101) = 4$$
$$\text{AdjBC}(010101010) = 0$$

Write a program which takes as input integers $n$ and $k$ and returns the number of bit strings $x$ of $n$ bits (out of $2^n$) that satisfy $\text{AdjBC}(x) = k$. For example, for 5 bit strings, there are 6 ways of getting $\text{AdjBC}(x) = 2$:

$$11100, 01110, 00111, 10111, 11101, 11011$$

**Input**

The first line of input contains a single integer $P$, $(1 \leq P \leq 1000)$, which is the number of data sets that follow. Each data set is a single line that contains the data set number, followed by a space, followed by a decimal integer giving the number ($n$) of bits in the bit strings, followed by a single space, followed by a decimal integer ($k$) giving the desired adjacent bit count. The number of bits ($n$) will not be greater than 100 and the parameters $n$ and $k$ will be chosen so that the result will fit in a signed 32-bit integer.

**Output**

For each data set there is one line of output. It contains the data set number followed by a single space, followed by the number of $n$-bit strings with adjacent bit count equal to $k$.

<table>
<thead>
<tr>
<th>Sample Input</th>
<th>Sample Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 1 5 2 2 30 4 5 60 7 8 9 90 10</td>
<td>1 6 2 63426 3 1861225 4 168212501 5 44874764 6 160916 7 22937308 8 99167 9 15476 10 23076518</td>
</tr>
</tbody>
</table>