We are familiar with the Fibonacci sequence (1, 1, 2, 3, 5, 8, ...). What if we define a similar sequence for strings? Sounds interesting? Let's see.

We define the follwing sequence:

$BFS(0) = 0$ $BFS(1) = 1$ (here "0" and "1" are strings, not simply the numerical digit, 0 or 1)

for all $(n > 1)$ $BFS(n) = BFS(n-2) + BFS(n-1)$ (here, '+' denotes the string concatenation operation). (i.e. the n-th string in this sequence is a concatenation of a previous two strings).

So, the first few strings of this sequence are: 0, 1, 01, 101, 01101, and so on.

Your task is to find the N-th string of the sequence and print all of its characters from the i-th to j-th position, inclusive. (All of N, i, j are 0-based indices)

Input

The first line of the input file contains an integer T ($T \leq 100$) which denotes the total number of test cases. The description of each test case is given below:

Three integers N, i, j ($0 \leq N, i, j \leq 2^{31} - 1$) and ($i \leq j$ and $j - i \leq 10000$). You can assume that, both i and j will be valid indices (i.e. $0 \leq i, j < length\ of\ BFS(N)$).

Output

For each test case, print the substring from the i-th to the j-th position of $BFS(N)$ in a single line.

Sample Input

```
3
3 1 2
1 0 0
9 5 12
```

Sample Output

```
01
1
10101101
```