F - Coherence

Input: `standard input`
Output: `standard output`

We want to display a two-color pattern on a rectangular array of pixels, with k of the pixels in the foreground color. We want to choose the pattern so that it minimizes the boundary between the foreground and the background.

The length of the boundary is the number of vertical and horizontal pixel edges that separate a foreground pixel from a background pixel. For example, this picture shows a rectangular array with three rows and six columns that has 5 foreground pixels (indicated by '`X`'). The boundary in this case has length equal to 6: the upper left `X` is adjacent to 1 background pixel, the upper middle `X` is adjacent to 1, the rightmost `X` is adjacent to 3, the lower left `X` is adjacent to 0, and the lower right `X` is adjacent to 1.

```
- - - - - -
X X X - - -
X X - - - -
```

Given the height of the array, the width of the array, and k (the number of foreground pixels), you must find the length of the minimum possible boundary.

Input

On the first line one positive number: the number of testcases. Each test case is defined in a line that contains three integer values $numRows$, $numCols$, and k ($1 \leq numRows, numCols \leq 30$, $0 \leq k \leq numRows * numCols$), representing (respectively) the height of the array, the width of the array, and the number of foreground pixels.

Output

Per testcase print one line with the length of the minimum possible boundary.

Sample Input

```
3
6 6 5
4 6 16
9 5 0
```

Sample Output

```
5
4
```

0