



CUTTING A GRID

There is a rectangular grid of c columns by r rows of squares. The squares of the grid are colored either black or white, with b of them being black. The squares in the four corners of the grid are always white.

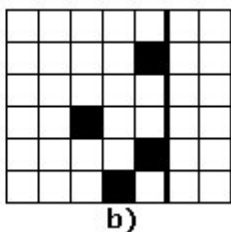
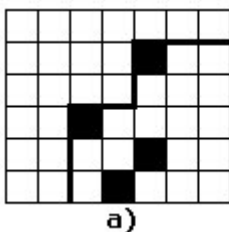
The grid must be cut into two pieces. The rules for cutting are as follows:

- Cuts must follow the lines in the grid.
- A cut must start somewhere on the bottom or the left side of the grid.
- A cut must proceed up or right (never left or down).
- A cut is complete when it reaches either the top or the right side of the grid.

The *area* of each of the pieces is the number of squares on that piece. A *turn* is every time the direction of the cut changes, that is, you were cutting up and changed to cutting right or vice versa.

Cut the grid so that:

- All the black squares are on the same piece.
- The area of the white piece is maximized.
- The number of turns you made while cutting does not exceed k , where k is a given integer.



Example of two different ways of cutting the same grid, figure a) leaves all the black squares on the right piece using 3 turns while figure b) leaves them on the left piece using 0 turns. Figure a) represents the best solution for this grid when $k=4$ because the area of the white piece is 21 and the number of turns does not exceed 4.

TASK

Write a program that determines how to cut the grid without exceeding the number of turns so that the white piece has maximal area.

You can be sure that it will always be possible to cut the grid in two pieces following the rules above.

CONSTRAINTS

$0 < c \leq 5\,000$
 $0 < r \leq 5\,000$
 $0 < b \leq 5\,000$
 $0 < k \leq 1\,000$



INPUT

Your program read from the file `cutting.in` the following data.

cutting.in	DESCRIPTION
7 6 4 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0	LINE 1: Contains 3 space-separated integers, respectively: c , r and k . NEXT r LINES: Each line contains c space-separated integers that represent the c squares of that row in the grid. A 0(zero) represents a white square and a 1 (one) represents a black square.

OUTPUT

Your program must write to the file `cutting.out` the following data.

cutting.out	DESCRIPTION
21	LINE 1: Must contain one integer representing the maximum possible area for the white piece.

GRADING INFO

For one test case, every test run will satisfy the following constraints:

$$0 < c \leq 10$$

$$0 < r \leq 10$$

$$0 < b \leq 50$$

$$0 < k \leq 5$$