

3702 ASCII Art

ASCII art is an art of creating pictures with a grid of ASCII characters. There are many styles of ASCII art, but we are interested in the most primitive one, where just an overall character density is used to represent differently shaded areas of the picture.

You should write a proof-of-concept program that renders a filled closed polygon with a rectangular grid of ASCII characters. The whole process is explained in detail below.

Let OXY be a Cartesian coordinate system with OX pointing to the right and OY pointing up. Drawing canvas is bounded with $(0, 0) - (w, h)$ rectangle. Pixels on the canvas are $(x, y) - (x + 1, y + 1)$ squares where x and y are integers such that $0 \leq x < w$ and $0 \leq y < h$. A filled closed polygon without selfintersections and self-touchings (but not necessarily convex) is drawn on the canvas. Pixels of the canvas become partially filled during the process. Each pixel is represented by an ASCII character depending on the percentage of its filled area according to the following table:

<i>Pixel percentage area filled</i>	<i>Character name</i>	<i>Glyph</i>	<i>ASCII code</i>
From 0% inclusive to 25% exclusive	Full stop	.	46
From 25% inclusive to 50% exclusive	Plus sign	+	43
From 50% inclusive to 75% exclusive	Small letter o	o	111
From 75% inclusive to 100% exclusive	Dollar sign	\$	36
100%	Number sign	#	35

The resulting ASCII characters for all pixels are printed top-to-bottom and left-to-right to get a visual representation of the drawing.

Input

The input will contain several test cases, each of them as described below. Consecutive test cases are separated by a single blank line.

The first line of the input contains integers n , w , and h ($3 \leq n \leq 100, 1 \leq w, h \leq 100$) — number of vertices in the polygon, width and height of the canvas respectively. The following n lines contain coordinates of the polygon vertices in clockwise order. Point i is described by two integers x_i and y_i ($0 \leq x_i \leq w, 0 \leq y_i \leq h$).

Output

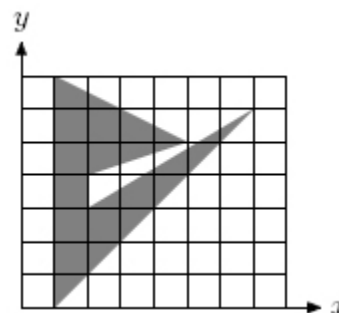
For each test case, write to the output h lines with w ASCII characters each that represent ASCII art drawing of the given polygon.

The outputs of two consecutive cases will be separated by a blank line.

Note: The picture on the right corresponds to the sample below.

Sample Input

```
6 8 7
7 6
1 0
1 7
5 5
```



2 4

2 3

Sample Output

```
.$+.....  
.##$+...  
.$$oo+..  
.#+$o...  
.##o....  
.#o.....  
.o.....
```