

# A

# ASCII Diamond

Input: Standard Input  
Output: Standard Output

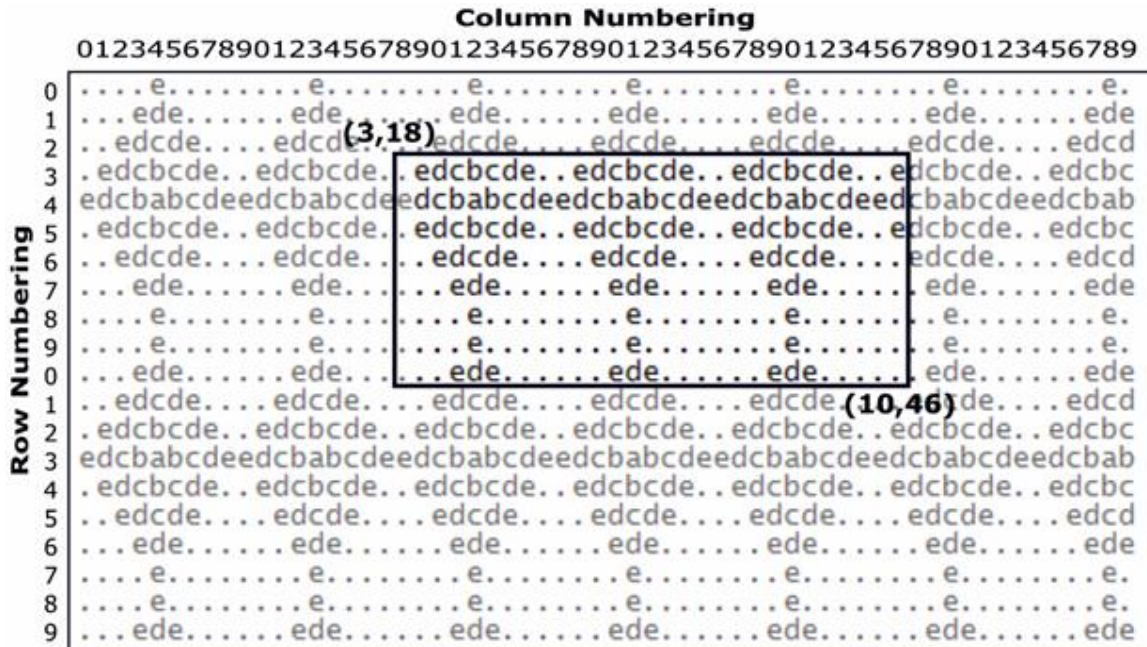


ASCII diamonds can be drawn with integer side lengths. Each layer of this diamond is drawn with a single ASCII alphabet, starting with 'a' and ending with 'z' (starting from the center) and continues in cyclic order.

a	.b.	..c..	...d...	....e....	.....f.....	.....g.....	.....h.....
	bab	.cbc.	..dcd..	...ede...	....fef....	.....gfg....	.....hgh....
	.b.	cbabc	..dcbcd.	...edcde..	....fedef...	.....gfefg...	.....hgfgh....
		.cbc.	dcbabcd	..edcbde.	...fedcdef..	....gfedefg...	.....hgfefgh....
		..c..	..dcbcd.	edcbabcde	..fedcbdef..	...gfedcdefg..	....hgfedefgh...
			..dcd..	..edcbde.	fedcbabcdef	gfedcbcddefg	hgfedcdefgh..
			...d...	...edcde..	..fedcbdef.	gfedcbabcdefg	hgfedcbcddefgh.
				...ede...	...fedcdef..	..gfedcbcddefg.	hgfedcbabcdefgh
				....e....	....fedef...	...gfedcdefg..	hgfedcbcddefgh.
					....fef....	....gfedefg...	hgfedcdefgh..
					.....f.....	.....gfefg....	hgfedefgh...
						.....gfg.....	hgfefgh....
						.....g.....	hgfgh....
							hgh....
							.....h.....
N=1	N=2	N=3	N=4	N=5	N=6	N=7	N=8

Figure 1: ASCII diamond for different side lengths.

Any one of these **ASCII** diamonds can be used to draw an infinite plane by using this as a tile. For example **ASCII** diamond of length 5 can be used to draw such an infinite grid. Only first 20 row and 60 columns are shown below:



Here rows and columns are numbered starting from zero. By specifying the topmost row (**row<sub>1</sub>**), leftmost column (**col<sub>1</sub>**), bottommost row (**row<sub>2</sub>**) and rightmost column (**col<sub>2</sub>**) we can specify a portion of such an infinite grid (also shown in figure above).

Given the side length of the tile to be used, the topmost row (**row<sub>1</sub>**), leftmost column (**col<sub>1</sub>**), bottommost row (**row<sub>2</sub>**) and rightmost column (**col<sub>2</sub>**) you have to print the pattern within these four boundaries (inclusive).

## Input

Input contains at most **125** sets of inputs. But not all cases are extreme.

Each set of input contains five integers: **N** ( $0 < N \leq 20000$ ), **row<sub>1</sub>**, **col<sub>1</sub>**, **row<sub>2</sub>**, **col<sub>2</sub>** ( $0 \leq \text{row}_1 \leq \text{row}_2 \leq 20000$ ,  $0 \leq \text{col}_1 \leq \text{col}_2 \leq 20000$ ,  $0 \leq (\text{row}_2 - \text{row}_1 + 1) * (\text{col}_2 - \text{col}_1 + 1) \leq 40000$ ). Here **N** denotes that the side length of the tiles used to draw the plane should be **N**. The meaning of **row<sub>1</sub>**, **col<sub>1</sub>**, **row<sub>2</sub>**, **col<sub>2</sub>** are given in the problem statement. The first sample input corresponds to the figure above.

Input is terminated by a line where the first integer is zero.

## Output

For each line of input produce **(row<sub>2</sub> - row<sub>1</sub> + 2)** lines of output. First line contains serial of output. Each of the next lines contain **(col<sub>2</sub> - col<sub>1</sub> + 1)** characters. These lines describe the patterns within the specified rows and columns. Look at the output for sample input for details. The output file size is less than **1 MB**.

### Sample Input

```
5 3 18 10 46
100 50 50 69 69
0 2 3 4 5
```

### Output for Sample Input

```
Case 1:
.edcbcbde..edcbcbde..edcbcbde..e
edcbabcbdeedcbabcbdeedcbabcbdeed
.edcbcbde..edcbcbde..edcbcbde..e
..edcbde...edcbde...edcbde...
...ede.....ede.....ede.....
....e.....e.....e.....
....e.....e.....e.....
...ede.....ede.....ede.....
Case 2:
utsrqponmlkjihgfedcb
tsrqponmlkjihgfedcba
srqponmlkjihgfedcbaz
rqponmlkjihgfedcbazy
qponmlkjihgfedcbazyx
ponmlkjihgfedcbazyxw
onmlkjihgfedcbazyxwv
nmlkjihgfedcbazyxwvu
mlkjihgfedcbazyxwvut
lkjihgfedcbazyxwvuts
kjihgfedcbazyxwvutsr
jihgfedcbazyxwvutsrq
ihgfedcbazyxwvutsrqp
hgfedcbazyxwvutsrqp
gfedcbazyxwvutsrqp
fedcbazyxwvutsrqpnm
edcbazyxwvutsrqpnm
dcbazyxwvutsrqpnmkl
cbazyxwvutsrqpnmklj
bazyxwvutsrqpnmklji
```

---

Problem setter: Shahriar Manzoor, Special Thanks: Sohel Hafiz, Md. Arifuzzaman Arif