Consider an infinite square grid with a clockwise spiral of consecutive positive integers. Number 1 is placed at the center, with 2 at its right, 3 bellow 2, and so one, and so forth. Having placed all number from 1 to $n-1$, n is placed in the same line with $n-1$ and $n-2$ unless the cell to the right of the $n-1$, in the $[n-2, n-1]$ direction, is empty, in wich case n is placed in this cell. The central 11×11 square of the spiral is shown in the figure bellow.

111	112	113	114	115	116	117	118	119	120	121
110	73	74	75	76	77	78	79	80	81	82
109	72	43	44	45	46	47	48	49	50	83
108	71	42	21	22	23	24	25	26	51	84
107	70	41	20	7	8	9	10	27	52	85
106	69	40	19	6	1	2	11	28	53	86
105	68	39	18	5	4	3	12	29	54	87
104	67	38	17	16	15	14	13	30	55	88
103	66	37	36	35	34	33	32	31	56	89
102	65	64	63	62	61	60	59	58	57	90
101	100	99	98	97	96	95	94	93	92	91

The spiral of numbers has some intriguing features: a lot of prime numbers form diagonal lines in the spiral. This is the case of 3, 13 31, 57 and 91 in the southeast diagonal and is also the case of 5, 17 and 37 in the southwest diagonal. As you would expect this is not a general rule since 65, the next number of southwest diagonal is not a prime number.

Nevertheless the spiral is worth a little more investigation and we would like you to write a program that given a positive integer n returns its neighboring numbers in the spiral, i.e. a 3×3 square with n in the center, surrounded by the numbers that are placed in the same relative positions in the spiral.

Input

The input file contains several test cases, each of them is a positive integer less than 2^{30} (a 4 byte integer) on a line by itself.

Output

For each test case, your program must write the neighboring numbers of the input number in 3 lines. Each line has 3 integers separated by semicolons. There must also be a semicolon at the end of each line.

Sample Input

```
11
```

Sample Output

```
9;10;27;
2;11;28;
3;12;29;
```