Actividad 1.3

Programación Avanzada

Instrucciones para envío en Blackboard

Guarda tus archivos con el nombre según la siguiente regla:

- M<matrícula>pX.cpp, donde <matrícula>corresponde a los 6 dígitos de su mátrícula UDEM y X al número de programa. Ejemplo: Si tu matrícula es 123456, el archivo para el problema 2, se deberá llamar M123456p2.cpp o .java
- Incluye el Código de Honor en cada programa como comentario.
- Si se realizan los programas en pareja, sólo un estudiante deberá subir el archivo nombrado con su matrícula. Importante poner **ambas matrículas y nombres** en el código fuente y/o en los comentarios al subir los programas en Blackboard.
- **Nota: Sino sigues esta regla, tus programas NO serán calificados.

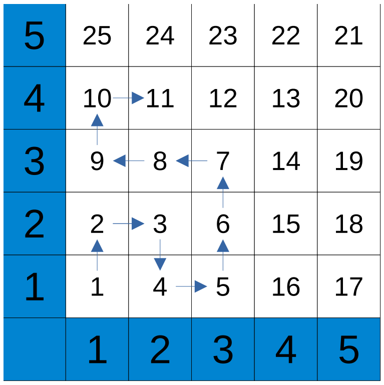
Resolver los siguientes problemas de programación.

Problem 1. *Walking Roger*

Standard Input

Time limit: 3 seconds

One day, Roger dreamed that he was walking around an $N \times N$ board. His goal was to go through all the squares once. So his strategy was to walk like a snake starting from position (1,1). Observe the next figure of a tour for a 5×5 board.



5	25	24	23	22	21
4	10	11	12	13	20
3	9	8	7	14	19
2	2	3	6	15	18
1	1	4	5	16	17
	1	2	3	4	5

Roger's speed is 1 second per square. Therefore, at second 9 he was at square (1,3) and at second 17 at (5,1).

Your job is to determine where Roger was at given number of seconds, or at what second he arrived at given position. Consider that the board is huge and that Roger walked until 10^{10} seconds.

Input

Input consists of several test cases. For each case there is a line containing either one or two integer numbers. If there is one number s, it indicates the number of seconds. If there are two numbers x and y, they are separated by a single space and indicate a position in the board.

Output

For each case, you must print either a line with two integers x and y (a position in the board) separated by a single space, or a line with one integer s (the number of seconds).

Example Input

```
9
17
3 3
1 5
1
0
```

Example Output

```
1 3
5 1
7
25
1 1
```

Problem 2. *Friends Primes*

Standard input
Time limit: 3 seconds

Friends Primes are 3 or more consecutive and equidistant primes. For example 3,5, and 7 are Friends Primes, because they are consecutive and the distance between them is 2.

You have to write a program that prints all the Friends Primes sets between two integers (both inclusive).

Input

The input consists of several test cases. Each case have two positive integers a and b, $0 \leq a, b \leq 10^6$. The entry ends with an entry with two zeros.

Output

For each case, print all the Friends Primes sets between a and b. Numbers on each set must be separated by a single space, each set must be on a different line.

***Important*: Do not print a subset of a serie. For example, a series of four primes, all 4 must be in the interval or they must not be printed.

Sample Input

```
1 100
2 8
190 265
0 0
```

Sample Output

```
3 5 7
47 53 59
3 5 7
199 211 223
```