

# Imagen pareja

Enlace: <https://codeforces.com/contest/1368/problem/C>

Leo hace dibujos en su libreta cuadriculada. Asumiremos que las hojas de su libreta son infinitas en todas las direcciones. Leo no es un gran dibujante y solo hace dibujos pintando algunas de las celdas de color gris con su lápiz.

Consideramos que dos celdas son adyacentes si tienen en común una línea que las separa. Llamamos vecinos a las 4 celdas que son adyacentes.

A Leo uno de sus dibujos le resulta interesante si las siguientes condiciones se cumplen:

- El dibujo es **conexo**, es decir, que de cualquier celda gris puedes llegar a todas las demás, moviéndote solo a celdas adyacentes en cada paso.
- Todas las **celdas grises** tienen un **número par** de **vecinos grises**.
- Hay exactamente  $n$  **celdas grises** con **todos** sus **vecinos grises**.

Leo no consigue hacer dibujos interesantes para algunos valores de  $n$ . Podrías ayudarlo, diciendo qué celdas debería pintar?

## Input Format

La entrada consiste en un único entero  $n$ .

## Constraints

$$1 \leq n \leq 500$$

## Output format

En la primera línea imprime un entero  $k$  — el número de celdas grises en el dibujo.  $k \geq 1$  y, por razones técnicas,  $k \leq 5 \cdot 10^5$

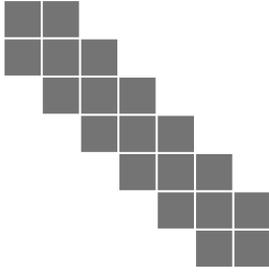
Las siguientes  $k$  líneas deben contener dos enteros  $x, y$  — las coordenadas de las celdas grises. No debe de repetirse ninguna celda y  $-10^9 \leq x, y \leq 10^9$

Siempre se puede encontrar una solución al problema con números suficientemente pequeños, es decir, sin incumplir estas restricciones.

En el problema original podréis encontrar un ejemplo para  $n = 4$ , con la solución y una imagen de esta.

## Solución

Existen muchas posibles soluciones, una de las más simples consiste en el siguiente patrón:



Este patrón usa  $3n + 4$  cuadrados grises, y resulta muy sencillo de programar.

## Código

### C++

```
1  #include <bits/stdc++.h>
2  using namespace std;
3
4  int main() {
5      int n;
6      cin >> n;
7      cout << 3*n+4 << endl;
8      cout << 0 << ' ' << 0 << endl;
9      for (int i = 0; i <= n; ++i) {
10         cout << i+1 << ' ' << i << endl;
11         cout << i << ' ' << i+1 << endl;
12         cout << i+1 << ' ' << i+1 << endl;
13     }
14 }
```

### Python

```
1  n = int(input())
2  print(3*n+4)
3  print(0, 0)
4  for i in range(0, n+1):
5      print(i+1, i)
6      print(i, i+1)
7      print(i+1, i+1)
```