ANÁLISIS Y DISEÑO DE ALGORITMOS
Ing. Román Martínez M.
Semestre Agosto-Diciembre 2016

TAREA #8

En esta tarea te adentrarás en la comprensión de la técnica de los algoritmos voraces, a través de los problemas del árbol de extensión mínima y del camino más corto, con los algoritmos de **Prim** y **Dijkstra**.

El primer paso de la tarea es que comprendas perfectamente la explicación de la propuesta de implementación del algoritmo de Dijkstra que se encuentra en la siguiente liga: https://mix.office.com/watch/l6o44fm9rvi4. Como evidencia de esta comprensión y de tus habilidades de programación, deberás implementar en C++ esta versión del algoritmo. Dado que es un algoritmo clásico, hay una gran cantidad de implementaciones que se pueden encontrar en Internet, así que NO caigas en la tentación de buscar un código ya hecho y copiarlo, pues ese no es el objetivo. En esta tarea, se busca que dada esta descripción del algoritmo, tú realices la implementación desde cero. El programa que codifiques deberá pedirle al usuario el nodo vértice a partir del cual se desean tener los caminos más cortos, y dará como resultado, los arcos que conforman la solución, junto con el costo que se tiene para llegar a cada nodo en el camino más corto. Prueba tu algoritmo con algún caso que tú mismo diseñes, y una vez que estés seguro que funciona, pruébalo con los 2 casos de grafos que se anexan a esta tarea.

El segundo paso consiste en implementar el algoritmo de Prim para obtener el árbol de extensión mínima. Como podrás notarlo, la estructura del algoritmo de Dijkstra sirve tal cual para implementar el algoritmo de Prim, dado que el proceso de selección de nodos es prácticamente igual, pero sin considerar los costos acumulados de un camino. Modifica el algoritmo de Disjktra para tener rápidamente el algoritmo de Prim implementado. El programa que codifiques no necesita pedirle datos al usuario, sino simplemente procesar el grafo que se indique y mostrar como resultado el conjunto de arcos que conforman la solución, así como el costo total de conexión con estos arcos. Prueba tu algoritmo con algún caso que tu mismo diseñes, y una vez que estés seguro que funciona, pruébalo con los 2 casos de grafos que se anexan a esta tarea.

Una vez que tengas implementados y funcionando los dos algoritmos, utilízalos para implementar un programa que resuelva el siguiente problema:

El alcalde de cierta pequeña ciudad, ha decidido generar ahorros a través de un uso restringido del alumbrado público en las calles. Todas las calles de la ciudad están alumbradas con arbotantes de luz que están colocados cada 10 metros, y el costo diario por la luz de cada arbotante es de 75 pesos. La estrategia de ahorro implicará sólo dejar prendidas las calles necesarias para para tener el máximo ahorro posible. Para decidir que calles dejar prendidas y cuales apagar sin afectar la seguridad de los habitantes que transitan por las noches, el alcalde pidió que en todos los cruces de calles se asegurara al menos una calle encendida, de tal manera que para fluir a cualquier lugar de la ciudad, hubiera un camino completo y completamente iluminado.
El alcalde te pide realizar un programa que sirva para lo siguiente:

1. Calcular el ahorro total diario que se tendría con esta estrategia.
2. Identificar caminos entre un cruce de calles y otro, que siendo los más eficientes (cortos), están completamente iluminados por las noches al tener esta estrategia de ahorro.

Para este caso, considera los siguientes detalles en tu implementación:

- El mapa con las calles de la ciudad será modelado en un grafo en donde los vértices son cruces de calles y los arcos son las distancias en metros que hay entre los dos cruces involucrados. Utiliza la primera matriz de adyacencias que se proporciona en el anexo, como dato fijo de entrada correspondiente a la modelación del mapa de la ciudad de la cual se necesitan obtener los resultados.
- La cantidad de arbotantes de luz es por segmento de calle entre cruce y cruce, y es independiente de los otros segmentos.
- Para el punto 2, el usuario del programa indicará a partir de que cruce de calles (identificado por un número) se desea obtener los caminos óptimos que si están iluminados por completo.

El reporte de esta tarea consiste en entregar una impresión del código de solución al problema presentado, documentando claramente la manera en que se utilizan los algoritmos implementados para solucionar el problema. Además deberás grabar un video y subirlo al espacio público de nuestro grupo en Facebook en el que muestres y expliques claramente la ejecución de tus programas y evidenciando su correcto funcionamiento. En este video comenta brevemente cómo fue tu proceso de solución y aprendizaje con esta tarea.

Para cualquier duda o detalle no especificado en este documento, consulta al profesor.

Esta tarea deberá entregarse a más tardar el **martes 1 de noviembre del 2016**.

ANEXO – Grafos para prueba

```
int g1[50][50] = {{0, 36, 47, 21, 23, 41, 38, 29, 18, 11, 28, 36, 47, 19, 26, 22, 34, 20, 25, 43,
12, 25, 27, 27, 31, 29, 17, 21, 32, 29, 27, 26, 16, 19, 39, 24, 48, 37, 34, 19, 19, 31, 38, 21, 16,
39, 28, 26, 26, 4}, {36, 0, 12, 44, 22, 30, 30, 36, 24, 24, 12, 17, 22, 20, 39, 34, 27, 27, 18, 22,
29, 15, 29, 23, 14, 20, 23, 24, 41, 26, 24, 20, 22, 26, 24, 37, 39, 19, 12, 11, 19, 4, 26, 40, 41,
11, 32, 21, 8, 28}, {47, 12, 0, 5, 36, 18, 32, 18, 26, 26, 32, 32, 27, 25, 12, 37, 20, 25, 13, 25,
4, 11, 25, 25, 15, 38, 30, 18, 36, 26, 23, 13, 14, 24, 42, 17, 36, 37, 19, 24, 39, 13, 33, 30, 17,
44, 34, 12, 31, 31}, {21, 44, 5, 0, 14, 9, 43, 36, 16, 5, 23, 23, 33, 29, 6, 23, 47, 45, 11, 39, 8,
48, 19, 38, 35, 15, 33, 23, 21, 29, 42, 18, 11, 37, 36, 25, 38, 42, 14, 31, 21, 27, 21, 12, 30, 37,
31, 35, 21, 40}, {23, 22, 36, 14, 0, 20, 3, 17, 24, 21, 31, 31, 34, 22, 33, 25, 22, 23, 41, 31, 27,
3, 17, 23, 16, 13, 23, 15, 12, 22, 23, 25, 18, 37, 33, 9, 14, 34, 29, 30, 27, 32, 24, 28, 13, 27,
31, 36, 24, 31}, {41, 30, 18, 9, 20, 0, 39, 19, 21, 11, 33, 31, 17, 28, 28, 31, 7, 11, 16, 17, 19,
27, 45, 37, 41, 23, 20, 16, 24, 26, 21, 16, 21, 26, 40, 26, 12, 44, 35, 16, 19, 15, 9, 18, 15, 40,
9, 23, 15, 42}, {38, 30, 32, 43, 3, 39, 0, 28, 11, 37, 22, 11, 3, 32, 46, 33, 28, 5, 23, 17, 22,
31, 24, 35, 40, 33, 32, 16, 44, 10, 21, 8, 40, 30, 32, 22, 31, 23, 33, 41, 28, 38, 39, 23, 12, 24,
21, 29, 26, 19}, {29, 36, 18, 36, 17, 19, 28, 0, 17, 21, 6, 31, 10, 9, 22, 34, 16, 20, 45, 12, 27,
27, 30, 27, 35, 24, 33, 30, 17, 15, 32, 26, 38, 41, 9, 20, 14, 16, 23, 28, 37, 23, 19, 35, 30, 20,
20, 32, 47, 48}, {18, 24, 26, 16, 24, 21, 11, 17, 0, 32, 34, 20, 12, 16, 29, 27, 8, 6, 17, 12, 4,
18, 30, 22, 45, 21, 28, 43, 14, 34, 32, 35, 35, 44, 19, 16, 25, 13, 17, 24, 19, 48, 41, 47, 34, 23,
37, 25, 40, 24}, {11, 24, 26, 5, 21, 11, 37, 21, 32, 0, 44, 9, 43, 31, 8, 17, 34, 21, 26, 26, 31,
31, 21, 22, 28, 22, 17, 27, 23, 34, 11, 20, 26, 33, 27, 24, 30, 28, 45, 24, 9, 26, 22, 33, 11, 24,
2, 40, 43, 44}, {28, 12, 32, 23, 31, 33, 22, 6, 34, 44, 0, 20, 24, 35, 2, 34, 19, 42, 35, 35, 18,
33, 35, 39, 35, 17, 22, 17, 24, 26, 12, 17, 24, 18, 23, 19, 28, 19, 30, 37, 24, 31, 32, 33, 26, 10,
13, 37, 31, 24}, {36, 17, 32, 23, 31, 31, 11, 31, 20, 9, 20, 0, 33, 38, 33, 22, 37, 26, 1, 44, 13,
42, 48, 33, 21, 40, 24, 27, 19, 16, 26, 26, 32, 19, 29, 19, 21, 31, 14, 34, 28, 19, 23, 47, 32, 16,
16, 8, 22, 26}, {47, 22, 27, 33, 34, 17, 3, 10, 12, 43, 24, 33, 0, 35, 14, 38, 25, 16, 21, 29, 22,
19, 41, 19, 45, 8, 21, 20, 36, 28, 38, 38, 35, 11, 16, 34, 50, 22, 11, 31, 8, 25, 14, 29, 35, 31,
15, 16, 31, 36}, {19, 20, 25, 29, 22, 28, 32, 9, 16, 31, 35, 38, 35, 0, 34, 48, 21, 24, 40, 25, 23,
46, 38, 38, 21, 33, 19, 39, 41, 26, 30, 33, 36, 41, 38, 6, 20, 12, 7, 29, 15, 40, 7, 41, 37, 19,
28, 31, 26, 28}, {26, 39, 12, 6, 33, 28, 46, 22, 29, 8, 2, 33, 14, 34, 0, 22, 26, 29, 35, 18, 32,
27, 22, 39, 34, 48, 28, 14, 23, 35, 43, 25, 20, 22, 19, 26, 39, 44, 36, 8, 26, 21, 21, 32, 28, 39,
29, 29, 39, 26}, {22, 34, 37, 23, 25, 31, 33, 34, 27, 17, 34, 22, 38, 48, 22, 0, 14, 30, 31, 29,
32, 48, 41, 29, 20, 29, 44, 2, 19, 23, 22, 41, 34, 30, 28, 37, 19, 15, 30, 32, 35, 43, 30, 26, 31,
5, 25, 12, 48, 5}, {34, 27, 20, 47, 22, 7, 28, 16, 8, 34, 19, 37, 25, 21, 26, 14, 0, 30, 47, 16,
27, 35, 27, 35, 21, 8, 14, 24, 10, 25, 41, 23, 19, 24, 20, 20, 43, 36, 25, 13, 22, 29, 44, 23, 23,
39, 33, 36, 32, 13}, {20, 27, 25, 45, 23, 11, 5, 20, 6, 21, 42, 26, 16, 24, 29, 30, 30, 0, 35, 32,
32, 26, 25, 38, 8, 30, 26, 31, 30, 34, 26, 44, 39, 27, 42, 31, 10, 34, 34, 29, 42, 41, 22, 39, 21,
44, 25, 13, 36, 7}, {25, 18, 13, 11, 41, 16, 23, 45, 17, 26, 35, 1, 21, 40, 35, 31, 47, 35, 0, 44,
36, 21, 9, 20, 27, 24, 33, 35, 25, 37, 25, 26, 41, 31, 16, 39, 35, 21, 40, 22, 22, 26, 24, 17, 9,
34, 34, 10, 21, 29}, {43, 22, 25, 39, 31, 17, 17, 12, 12, 26, 35, 44, 29, 25, 18, 29, 16, 32, 44,
0, 31, 45, 18, 28, 22, 37, 13, 40, 27, 40, 30, 17, 41, 33, 15, 40, 30, 36, 22, 23, 33, 17, 15, 32,
30, 21, 25, 43, 41, 22}, {12, 29, 4, 8, 27, 19, 22, 27, 4, 31, 18, 13, 22, 23, 32, 32, 27, 32, 36,
31, 0, 27, 18, 27, 26, 23, 7, 17, 11, 22, 22, 30, 34, 17, 37, 9, 23, 4, 8, 45, 43, 46, 22, 5, 10,
29, 34, 28, 26, 47}, {25, 15, 11, 48, 3, 27, 31, 27, 18, 31, 33, 42, 19, 46, 27, 48, 35, 26, 21,
45, 27, 0, 37, 26, 27, 11, 36, 24, 32, 32, 37, 28, 14, 8, 41, 24, 38, 26, 35, 31, 29, 27, 5, 34,
21, 16, 16, 15, 39, 23}, {27, 29, 25, 19, 17, 45, 24, 30, 30, 21, 35, 48, 41, 38, 22, 41, 27, 25,
9, 18, 18, 37, 0, 22, 26, 46, 21, 13, 10, 20, 25, 29, 34, 18, 28, 24, 13, 15, 33, 32, 40, 36, 14,
17, 19, 15, 28, 14, 31, 36}, {27, 23, 25, 38, 23, 37, 35, 27, 22, 22, 39, 33, 19, 38, 39, 29, 35,
38, 20, 28, 27, 26, 22, 0, 26, 15, 16, 35, 42, 26, 7, 16, 21, 28, 46, 18, 34, 46, 14, 26, 28, 32,
38, 46, 25, 14, 20, 33, 42, 6}, {31, 14, 15, 35, 16, 41, 40, 35, 45, 28, 35, 21, 45, 21, 34, 20,
21, 8, 27, 22, 26, 27, 26, 26, 0, 21, 31, 26, 15, 47, 13, 27, 34, 35, 9, 48, 30, 39, 22, 36, 29,
35, 22, 33, 27, 38, 33, 42, 40, 19}, {29, 20, 38, 15, 13, 23, 33, 24, 21, 22, 17, 40, 8, 33, 48,
29, 8, 30, 24, 37, 23, 11, 46, 15, 21, 0, 28, 3, 33, 21, 33, 11, 14, 27, 20, 41, 6, 20, 32, 46, 25,
35, 27, 27, 24, 20, 19, 22, 12, 31}, {17, 23, 30, 33, 23, 20, 32, 33, 28, 17, 22, 24, 21, 19, 28,
44, 14, 26, 33, 13, 7, 36, 21, 16, 31, 28, 0, 23, 21, 28, 12, 46, 34, 31, 45, 37, 26, 18, 35, 28,
16, 21, 25, 42, 34, 13, 18, 24, 17, 14}, {21, 24, 18, 23, 15, 16, 16, 30, 43, 27, 17, 27, 20, 39,
14, 2, 24, 31, 35, 40, 17, 24, 13, 35, 26, 3, 23, 0, 39, 22, 21, 16, 19, 31, 27, 41, 34, 24, 22,
29, 9, 23, 31, 37, 16, 33, 29, 27, 39, 25}, {32, 41, 36, 21, 12, 24, 44, 17, 14, 23, 24, 19, 36,
41, 23, 19, 10, 30, 25, 27, 11, 32, 10, 42, 15, 33, 21, 39, 0, 24, 45, 35, 21, 13, 22, 32, 24, 23,
30, 25, 40, 22, 11, 8, 28, 44, 30, 22, 16, 12}, {29, 26, 26, 29, 22, 26, 10, 15, 34, 34, 26, 16,
28, 26, 35, 23, 25, 34, 37, 40, 22, 32, 20, 26, 47, 21, 28, 22, 24, 0, 31, 6, 29, 31, 45, 21, 20,
33, 27, 38, 38, 31, 32, 24, 16, 14, 25, 29, 33, 18}, {27, 24, 23, 42, 23, 21, 21, 32, 32, 11, 12,
26, 38, 30, 43, 22, 41, 26, 25, 30, 22, 37, 25, 7, 13, 33, 12, 21, 45, 31, 0, 29, 23, 15, 15, 30,
21, 5, 32, 10, 17, 24, 21, 10, 40, 39, 29, 34, 29, 33}, {26, 20, 13, 18, 25, 16, 8, 26, 35, 20, 17,
26, 38, 33, 25, 41, 23, 44, 26, 17, 30, 28, 29, 16, 27, 11, 46, 16, 35, 6, 29, 0, 14, 8, 25, 20,
24, 25, 27, 11, 25, 12, 30, 28, 44, 21, 25, 38, 26, 24}, {16, 22, 14, 11, 18, 21, 40, 38, 35, 26,
24, 32, 35, 36, 20, 34, 19, 39, 41, 41, 34, 14, 34, 21, 34, 14, 34, 19, 21, 29, 23, 14, 0, 42, 30,
34, 27, 26, 17, 21, 12, 34, 21, 25, 22, 14, 32, 35, 47, 35}, {19, 26, 24, 37, 37, 26, 30, 41, 44,
33, 18, 19, 11, 41, 22, 30, 24, 27, 31, 33, 17, 8, 18, 28, 35, 27, 31, 31, 13, 31, 15, 8, 42, 0,
24, 45, 32, 32, 17, 19, 43, 20, 21, 25, 19, 10, 16, 27, 25, 18}, {39, 24, 42, 36, 33, 40, 32, 9,
19, 27, 23, 29, 16, 38, 19, 28, 20, 42, 16, 15, 37, 41, 28, 46, 9, 20, 45, 27, 22, 45, 15, 25, 30,
24, 0, 30, 8, 16, 33, 36, 33, 43, 33, 1, 28, 39, 36, 18, 25, 36}, {24, 37, 17, 25, 9, 26, 22, 20,
16, 24, 19, 19, 34, 6, 26, 37, 20, 31, 39, 40, 9, 24, 24, 18, 48, 41, 37, 41, 32, 21, 30, 20, 34,
45, 30, 0, 26, 41, 28, 26, 9, 30, 19, 19, 24, 14, 9, 25, 22, 33}, {48, 39, 36, 38, 14, 12, 31, 14,
25, 30, 28, 21, 50, 20, 39, 19, 43, 10, 35, 30, 23, 38, 13, 34, 30, 6, 26, 34, 24, 20, 21, 24, 27,
32, 8, 26, 0, 28, 40, 6, 33, 18, 20, 27, 42, 26, 30, 24, 38, 34}, {37, 19, 37, 42, 34, 44, 23, 16,
13, 28, 19, 31, 22, 12, 44, 15, 36, 34, 21, 36, 4, 26, 15, 46, 39, 20, 18, 24, 23, 33, 5, 25, 26,
32, 16, 41, 28, 0, 32, 29, 7, 23, 37, 9, 46, 32, 13, 26, 17, 22}, {34, 12, 19, 14, 29, 35, 33, 23,
17, 45, 30, 14, 11, 7, 36, 30, 25, 34, 40, 22, 8, 35, 33, 14, 22, 32, 35, 22, 30, 27, 32, 27, 17,
17, 33, 28, 40, 32, 0, 28, 42, 18, 17, 34, 29, 21, 25, 31, 32, 16}, {19, 11, 24, 31, 30, 16, 41,
28, 24, 24, 37, 34, 31, 29, 8, 32, 13, 29, 22, 23, 45, 31, 32, 26, 36, 46, 28, 29, 25, 38, 10, 11,
21, 19, 36, 26, 6, 29, 28, 0, 18, 5, 35, 24, 3, 29, 20, 41, 4, 26}, {19, 19, 39, 21, 27, 19, 28,
37, 19, 9, 24, 28, 8, 15, 26, 35, 22, 42, 22, 33, 43, 29, 40, 28, 29, 25, 16, 9, 40, 38, 17, 25,
12, 43, 33, 9, 33, 7, 42, 18, 0, 30, 35, 35, 17, 25, 27, 14, 20, 32}, {31, 4, 13, 27, 32, 15, 38,
23, 48, 26, 31, 19, 25, 40, 21, 43, 29, 41, 26, 17, 46, 27, 36, 32, 35, 35, 21, 23, 22, 31, 24, 12,
34, 20, 43, 30, 18, 23, 18, 5, 30, 0, 27, 22, 14, 47, 23, 13, 20, 21}, {38, 26, 33, 21, 24, 9, 39,
19, 41, 22, 32, 23, 14, 7, 21, 30, 44, 22, 24, 15, 22, 5, 14, 38, 22, 27, 25, 31, 11, 32, 21, 30,
21, 21, 33, 19, 20, 37, 17, 35, 35, 27, 0, 35, 32, 19, 34, 24, 13, 13}, {21, 40, 30, 12, 28, 18,
23, 35, 47, 33, 33, 47, 29, 41, 32, 26, 23, 39, 17, 32, 5, 34, 17, 46, 33, 27, 42, 37, 8, 24, 10,
28, 25, 25, 1, 19, 27, 9, 34, 24, 35, 22, 35, 0, 27, 21, 37, 23, 40, 30}, {16, 41, 17, 30, 13, 15,
12, 30, 34, 11, 26, 32, 35, 37, 28, 31, 23, 21, 9, 30, 10, 21, 19, 25, 27, 24, 34, 16, 28, 16, 40,
44, 22, 19, 28, 24, 42, 46, 29, 3, 17, 14, 32, 27, 0, 34, 22, 16, 19, 23}, {39, 11, 44, 37, 27, 40,
24, 20, 23, 24, 10, 16, 31, 19, 39, 5, 39, 44, 34, 21, 29, 16, 15, 14, 38, 20, 13, 33, 44, 14, 39,
21, 14, 10, 39, 14, 26, 32, 21, 29, 25, 47, 19, 21, 34, 0, 1, 36, 39, 28}, {28, 32, 34, 31, 31, 9,
21, 20, 37, 2, 13, 16, 15, 28, 29, 25, 33, 25, 34, 25, 34, 16, 28, 20, 33, 19, 18, 29, 30, 25, 29,
25, 32, 16, 36, 9, 30, 13, 25, 20, 27, 23, 34, 37, 22, 1, 0, 29, 17, 29}, {26, 21, 12, 35, 36, 23,
29, 32, 25, 40, 37, 8, 16, 31, 29, 12, 36, 13, 10, 43, 28, 15, 14, 33, 42, 22, 24, 27, 22, 29, 34,
38, 35, 27, 18, 25, 24, 26, 31, 41, 14, 13, 24, 23, 16, 36, 29, 0, 27, 45}, {26, 8, 31, 21, 24, 15,
26, 47, 40, 43, 31, 22, 31, 26, 39, 48, 32, 36, 21, 41, 26, 39, 31, 42, 40, 12, 17, 39, 16, 33, 29,
26, 47, 25, 25, 22, 38, 17, 32, 4, 20, 20, 13, 40, 19, 39, 17, 27, 0, 14}, {4, 28, 31, 40, 31, 42,
19, 48, 24, 44, 24, 26, 36, 28, 26, 5, 13, 7, 29, 22, 47, 23, 36, 6, 19, 31, 14, 25, 12, 18, 33,
24, 35, 18, 36, 33, 34, 22, 16, 26, 32, 21, 13, 30, 23, 28, 29, 45, 14, 0}};



int g2[50][50] = {{0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0}, {0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0,
1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,
0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0},
{0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0,
0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1,
0, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 1, 0, 1, 0, 0, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0, 0}, {0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 0}, {0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1}, {0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0,
0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0}};
```