

Федеральное государственное автономное образовательное учреждение высшего
образования

«Санкт-Петербургский национальный исследовательский университет
информационных технологий, механики и оптики»

Кафедра Вычислительной Техники

Задача №5
1401. “Игроки”

Выполнила: Калугина Марина
Группа: Р3202

г. Санкт-Петербург
2018-2019

В данной задаче необходимо заметить, что возможно собрать любой квадрат размера $2^n \times 2^n$ без одной клетки.

Каждый квадрат можно рекурсивно делить на 4 других квадрата (из-за того, что количество сторон - это степень двойки). В каждом квадрате этой рекурсии доску без одной клетки можно замостить данными нам фигурами. Т.к. по условию задачи на доске отсутствует одна клетка, то у нас остаются незакрытые 3 клетки из которых собирается фигура.

На каждом шаге рекурсии строю эту фигуру, которая собирается из “дырок”. Когда доходим до квадрата 2×2 и однозначно можем поставить фигуру - ставим ее.

Исходный код:

```
#include <iostream>
#include <cmath>

using namespace std;

int arr[512][512];

// i, j - координаты начала квадрата
// x, y - координаты дырки
// n - размер квадрата в текущей итерации
void gen(int x, int y, int i, int j, int n) {
    static int c; // счетчик для фигур
    c++;
    if ((x - j) < n / 2 && (y - i) < n / 2) {
        // cout << "четвертая четверть" << endl;
        if (n == 2) {
            arr[i][j + 1] = c;
            arr[i + 1][j] = c;
            arr[i + 1][j + 1] = c;
            return;
        }
        arr[i + n / 2 - 1][j + n / 2] = c;
        arr[i + n / 2][j + n / 2 - 1] = c;
        arr[i + n / 2][j + n / 2] = c;
        gen(x, y, i, j, n / 2);
        gen(j + n / 2, i + n / 2 - 1, i, j + n / 2, n / 2);
        gen(j + n / 2, i + n / 2, i + n / 2, j + n / 2, n / 2);
    }
```

```

        gen(j + n / 2 - 1, i + n / 2, i + n / 2, j, n / 2);
    } else if ((x - j) >= n / 2 && (y - i) < n / 2) {
//        cout << "первая четверть" << endl;
        if (n == 2) {
            arr[i][j] = c;
            arr[i + 1][j] = c;
            arr[i + 1][j + 1] = c;
            return;
        }
        arr[i + n / 2 - 1][j + n / 2 - 1] = c;
        arr[i + n / 2][j + n / 2 - 1] = c;
        arr[i + n / 2][j + n / 2] = c;
        gen(j + n / 2 - 1, i + n / 2 - 1, i, j, n / 2);
        gen(x, y, i, j + n / 2, n / 2);
        gen(j + n / 2, i + n / 2, i + n / 2, j + n / 2, n / 2);
        gen(j + n / 2 - 1, i + n / 2, i + n / 2, j, n / 2);
    } else if ((x - j) >= n / 2 && (y - i) >= n / 2) {
//        cout << "вторая четверть" << endl;
        if (n == 2) {
            arr[i][j] = c;
            arr[i + 1][j] = c;
            arr[i][j + 1] = c;
            return;
        }
        arr[i + n / 2 - 1][j + n / 2 - 1] = c;
        arr[i + n / 2][j + n / 2 - 1] = c;
        arr[i + n / 2 - 1][j + n / 2] = c;
        gen(j + n / 2 - 1, i + n / 2 - 1, i, j, n / 2);
        gen(j + n / 2, i + n / 2 - 1, i, j + n / 2, n / 2);
        gen(x, y, i + n / 2, j + n / 2, n / 2);
        gen(j + n / 2 - 1, i + n / 2, i + n / 2, j, n / 2);
    } else if ((x - j) < n / 2 && (y - i) >= n / 2) {
//        cout << "третья четверть" << endl;
        if (n == 2) {
            arr[i][j] = c;
            arr[i][j + 1] = c;
            arr[i + 1][j + 1] = c;
            return;
        }
        arr[i + n / 2 - 1][j + n / 2 - 1] = c;
        arr[i + n / 2][j + n / 2] = c;
        arr[i + n / 2 - 1][j + n / 2] = c;

```

```

        gen(j + n / 2 - 1, i + n / 2 - 1, i, j, n / 2);
        gen(j + n / 2, i + n / 2 - 1, i, j + n / 2, n / 2);
        gen(j + n / 2, i + n / 2, i + n / 2, j + n / 2, n / 2);
        gen(x, y, i + n / 2, j, n / 2);
    }
}

int main() {
    int n, x, y;
    cin >> n >> y >> x;
    x--;
    y--;
    n = (int) pow(2, n);
    if ((int) (pow(n, 2) - 1) % 3 != 0) {
        cout << -1;
        return 0;
    }
    gen(x, y, 0, 0, n);
    for (int i = 0; i < n; ++i) {
        for (int j = 0; j < n; ++j) {
            cout << arr[i][j] << " ";
        }
        cout << endl;
    }
    return 0;
}

```