1.Ý tưởng thực hiện:

- Ta thấy rằng để sinh ra xâu nhị phân tiếp theo mà theo thứ tự tăng dần thì đầu tiên ta phải tìm vị trí để tăng và tăng ít nhất có thể, ở đây dễ thấy là vị trí đầu tiên bằng 0 từ phải sang sẽ là vị trí cần tìm, sau khi tăng lên 1 thì để đảm bảo xâu kế tiếp là xâu nhỏ nhất thì những vị trí sau vị trí tìm sẽ bằng 0 để đảm bảo việc nhỏ nhất có thể

2. Minh hoạ hoạt động:

Ví dụ ta có xâu : 011

Ta tìm vị trí k là chữ số 0 đầu tiên tính từ bên phải của x: k = 1

Tăng x1 lên: x1 = 1

X2, x3 = 0

Vậy xâu tiếp theo là 100

3. biểu diễn thuật toán

- Giả sử cấu hình hiện tại x = x1x2..xn

Nếu xi = 1 với mọi i thì x là cấu hình cuối cùng, thuật toán liệt kê kết thúc

Gọi xk là chữ số 0 đầu tiên tính từ bên phải của x, như vậy x = x1x2…x(k-1)0111….1

Cấu hình tiếp theo y = y1y2…yn được tạo ra như sau

Yi = xi với 1 <= i <= k – 1, yi = 1 – x với k <= i <= n

Y = x1x2..x(k-1)10000….000

4. Độ phức tạp: O(n\*2^n)

5.Code

#include <bits/stdc++.h>

using namespace std;

int n, x[100];

void init();

void display();

void genNext();

bool isFinal();

//cu the

void init() {

    for (int i = 1; i <= n; i++) {

        x[i] = 0;

    }

}

void display() {

    for (int i = 1; i <= n; i++) {

        cout << x[i];

    }

    cout << endl;

}

bool isFinal() {

    for (int i = 1; i <= n; i++) {

        if (x[i] == 0) {

            return false;

        }

    }

    return true;

}

void genNext() {

    int t = n;

    while (x[t] == 1) {

        t--;

    }

    x[t] = 1;

    for (int i = t + 1; i <= n; i++) {

        x[i] = 0;

    }

}

int main () {

    #ifndef ONLINE\_JUDGE

    freopen("input.txt", "r", stdin);

    // freopen("output.txt", "w", stdout);

    #endif

    cin >> n;

    init();

    while (!isFinal()) {

        display();

        genNext();

    }

    display();

}