// A utility function generate all string without

// consecutive 1'sof size K

void generateAllStringsUtil(int K, char str[], int n)

{

    // print binary string without consecutive 1's

    if (n  == K)

    {

        // terminate binary string

        str[n] = '\0' ;

        cout << str << " ";

        return ;

    }

    // if previous character is '1' then we put

    // only 0 at end of string

    //example str = "01" then new string be "000"

    if (str[n-1] == '1')

    {

        str[n] = '0';

        generateAllStringsUtil (K , str , n+1);

    }

    // if previous character is '0' than we put

    // both '1' and '0' at end of string

    // example str = "00" then new  string "001" and "000"

    if (str[n-1] == '0')

    {

        str[n] = '0';

        generateAllStringsUtil(K, str, n+1);

        str[n] = '1';

        generateAllStringsUtil(K, str, n+1) ;

    }

}

// function generate all binary string without

// consecutive 1's

void generateAllStrings(int K )

{

    // Base case

    if (K <= 0)

        return ;

    // One by one stores every binary string of length K

    char str[K];

    // Generate all Binary string starts with '0'

    str[0] = '0' ;

    generateAllStringsUtil ( K , str , 1 ) ;

    // Generate all Binary string starts with '1'

    str[0] = '1' ;

    generateAllStringsUtil ( K , str , 1 );

}