

21168

DCCKM

D KM

KM

KM

KM

KM

M

M

```
#include <iostream>
#include <string>
#include <ctime>
#include <Windows.h>
#include <algorithm>
#include <vector>
#include <map>

using namespace std;

class NUMBER {
private:
    static vector<int> digit;
    static map<char, int> letter;
    static map<int, char> cryptographer;

    class COLORE {
    public:
        COLORE() {
            map<int, char> ::iterator it;
            for (auto i : digit) {
                it = cryptographer.find(i);
                letter.insert(map<char, int>::value_type(it->second, (i + 5) % 15));
            }
        }
    };

};

class SHOW {
public:
    SHOW() {
        HANDLE h; h = GetStdHandle(STD_OUTPUT_HANDLE);
        // Выводим исходное число и меняем фон за цифрами
        for (auto item : digit) {
            char symbol = cryptographer.find(item)->second; // Находим нужный
            символ по букве
        }
    }
};
```

```

        char color = letter.find(symbol)->second; // По букве находим
кодировку цвета
        SetConsoleTextAttribute(h, (((color << 4) | 15)));
        cout << item;
    }
    cout << "\n";

    // Выводим буквы в столбик, меняя их цвет
    for (int n = 0; n < 10; n++) {
        for (auto item : digit) {
            char symbol = cryptographer.find(item)->second;
            char color = letter.find(symbol)->second;
            SetConsoleTextAttribute(h, color);

            if (item > n) {
                cout << symbol;
            } else {
                cout << " ";
            }
        }
        cout << "\n";
    }
}

};

public:
    NUMBER() {
        int number = rand();
        while (number > 0) {
            digit.insert(digit.begin(), number % 10);
            number /= 10;
        }
        COLORE _COLORE;
        SHOW _SHOW;
    }
};

vector <int> NUMBER::digit;
map <char, int> NUMBER::letter;
map <int, char> NUMBER::cryptographer = {
    {0, 'W'}, {1, 'C'},
    {2, 'D'}, {3, 'F'},
    {4, 'G'}, {5, 'H'},
    {6, 'K'}, {7, 'L'},
    {8, 'M'}, {9, 'R'}
};

void main() {
    srand(time(NULL));
    setlocale(LC_ALL, "ru");
    NUMBER _NUMBER;
}

```