CompactPainter

Realizat de Boroica Marius și Ganea Alexandru-Gabriel

Colegiul Național Pedagogic “Regele Ferdinand”

Sighetu Marmației

Profesor coordonator :Negrea Beatrix

**Cuprins:**

Prezentarea generală a programului……………………………………………..pg.3

Aplicația folosită pentru convertirea caracterelor in pixeli (convert.cpp)………pg.9

CompactPainter.cpp……………………………………………………………..pg.11

Bibliografie……………………………………………………………………...pg.65

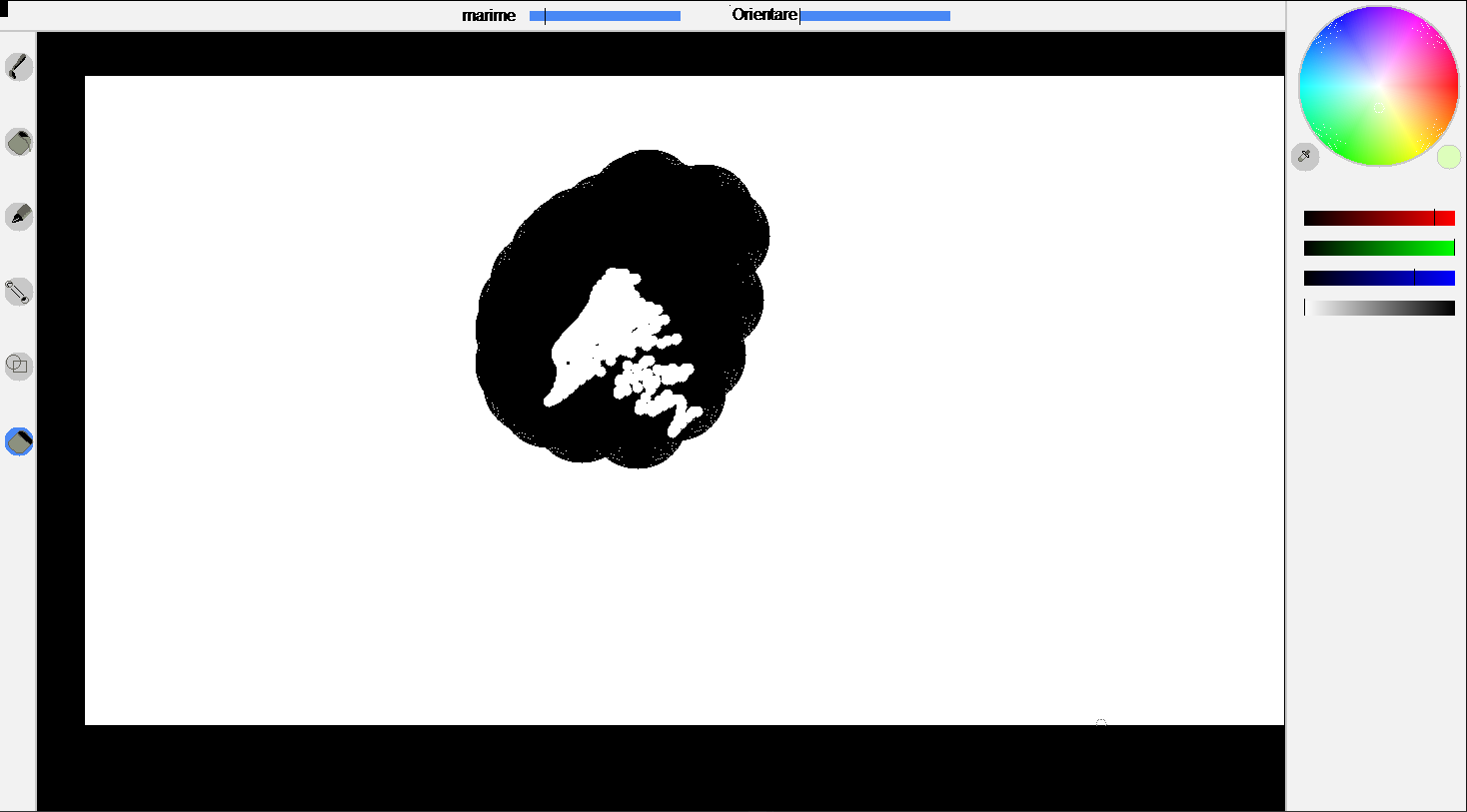
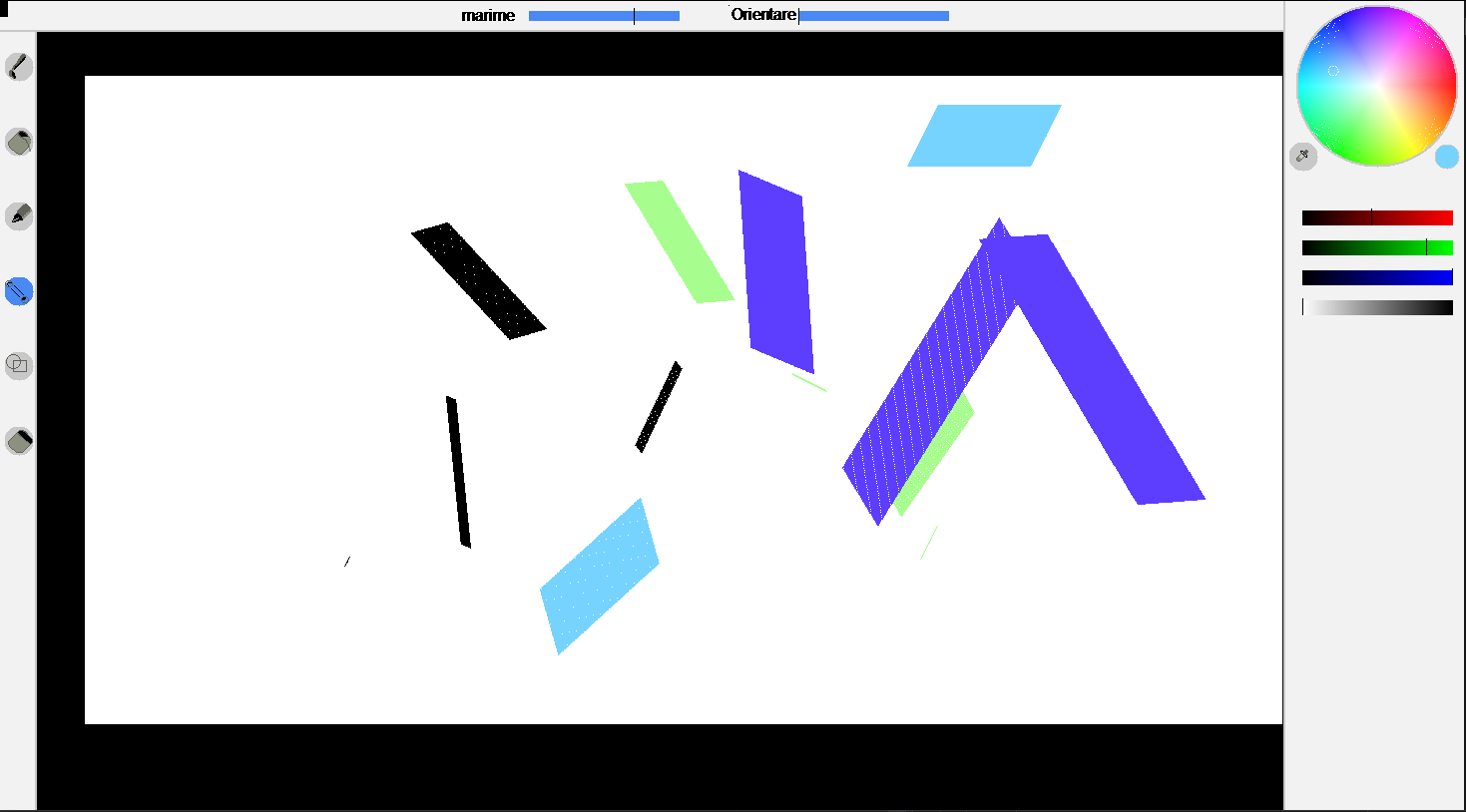
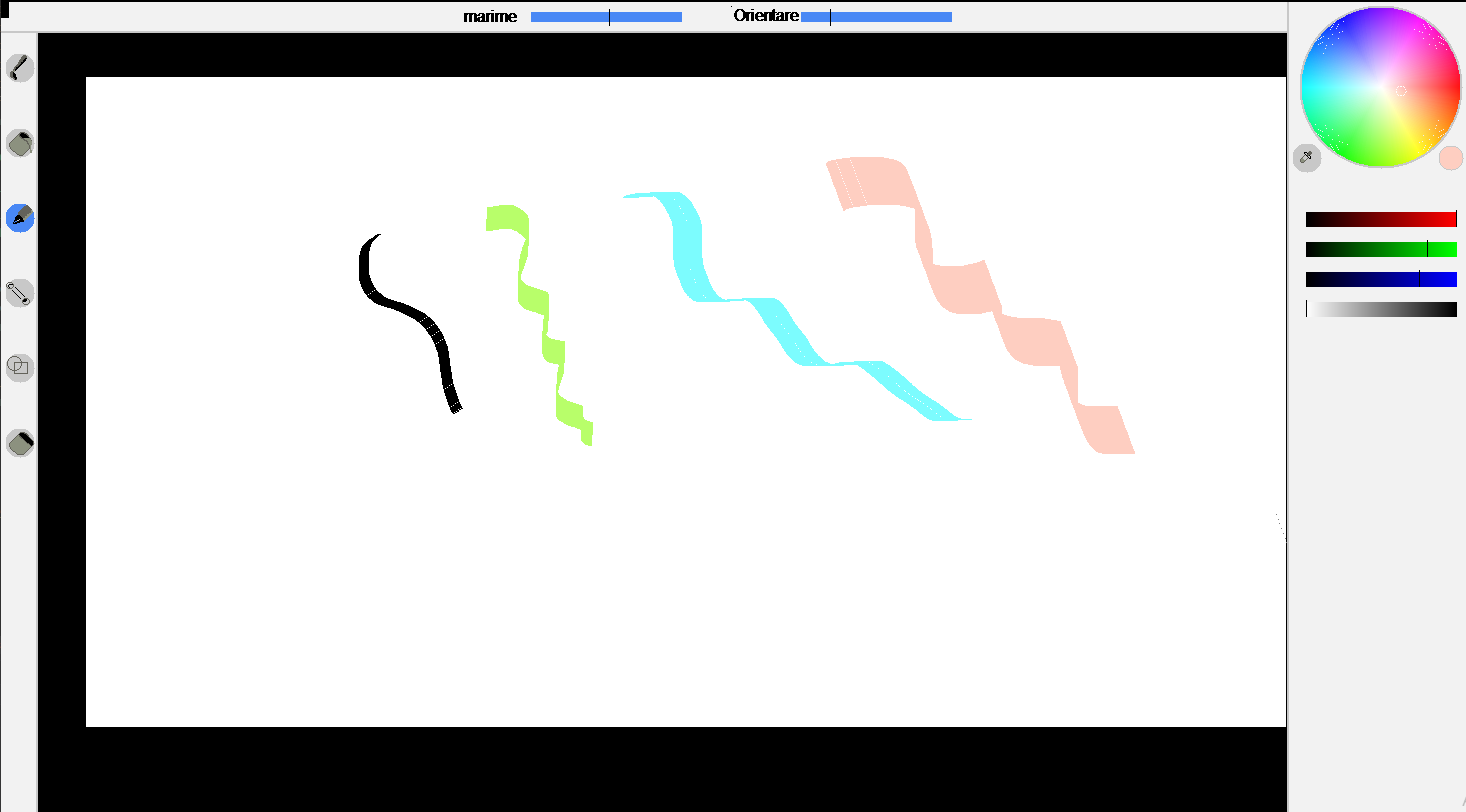
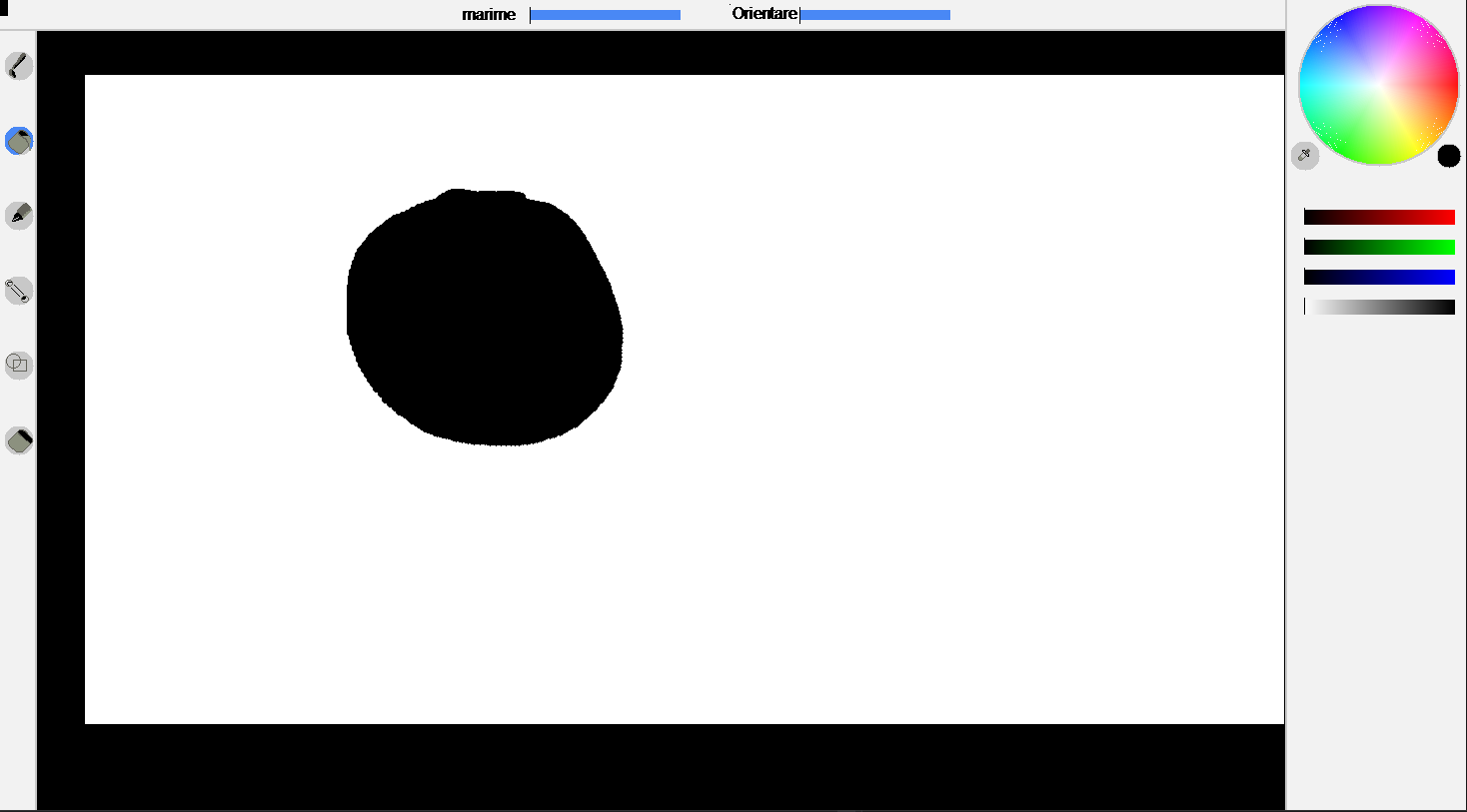
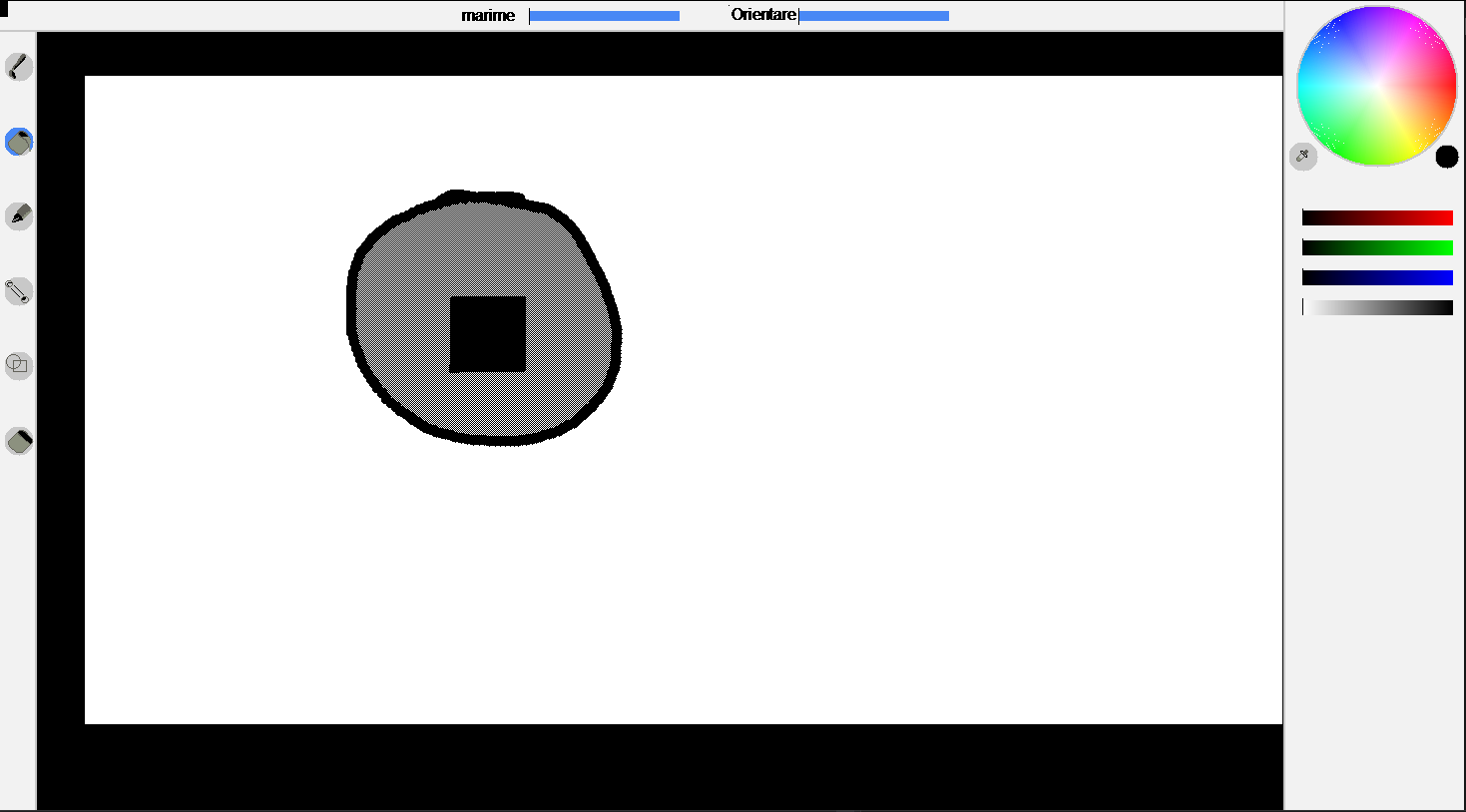
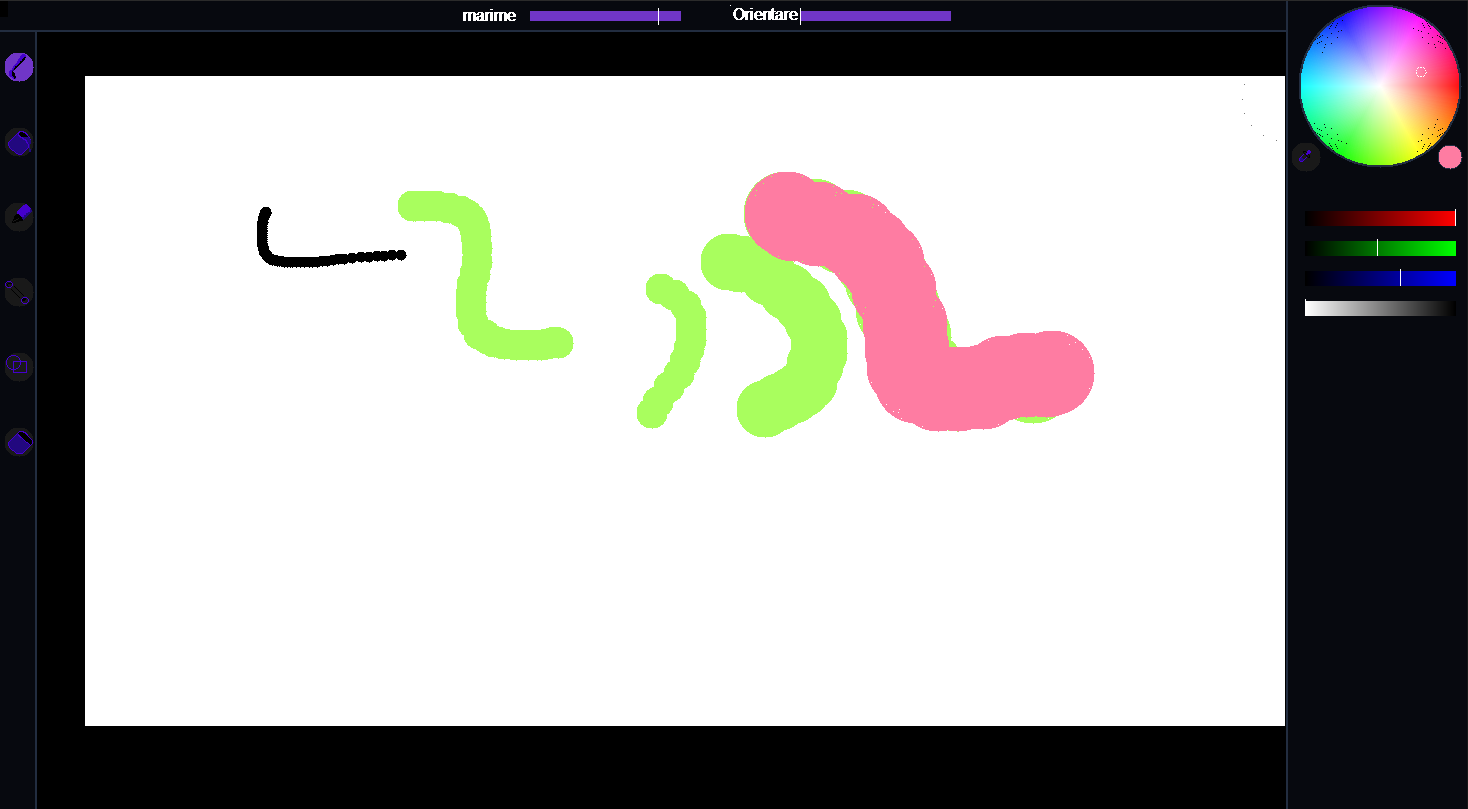
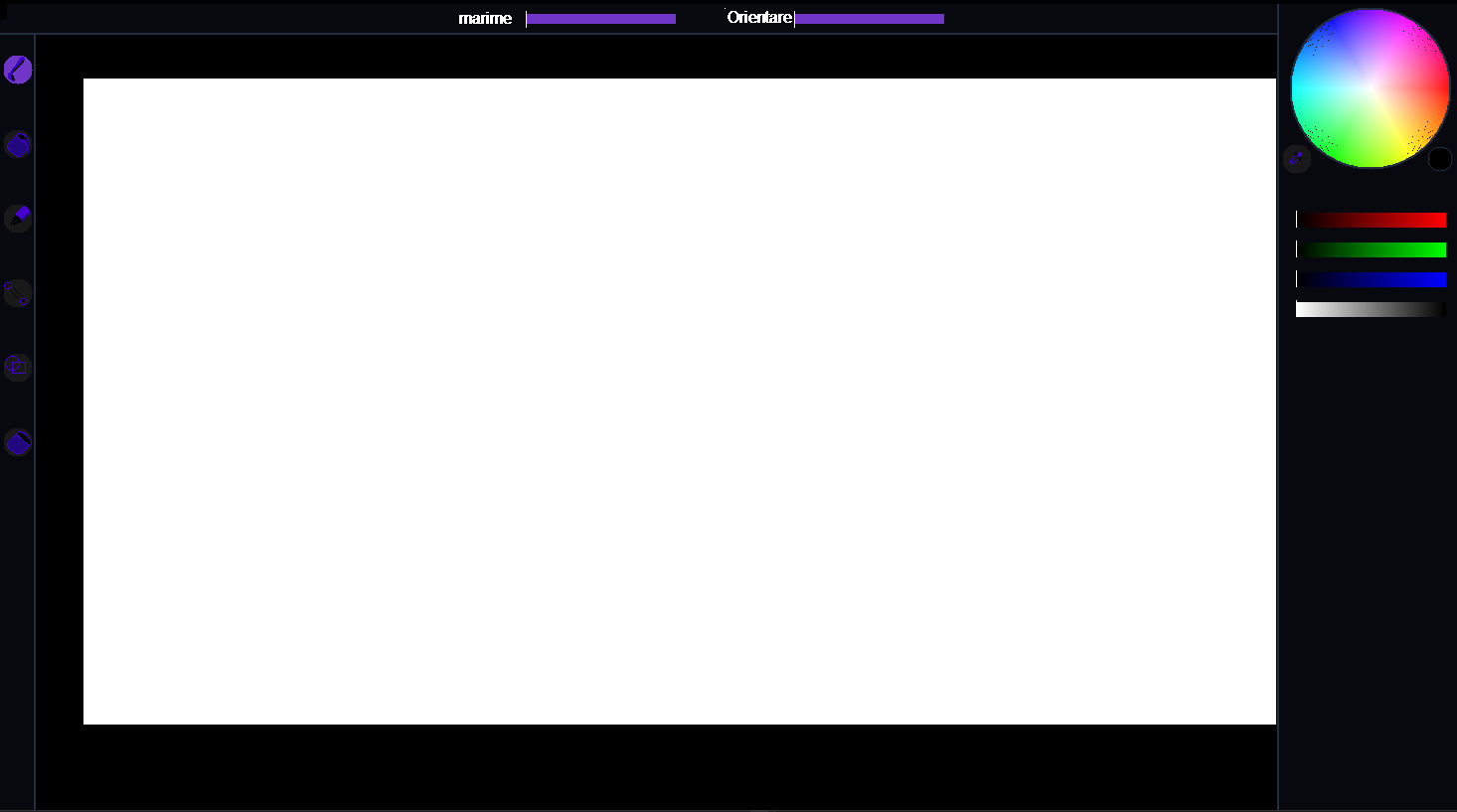
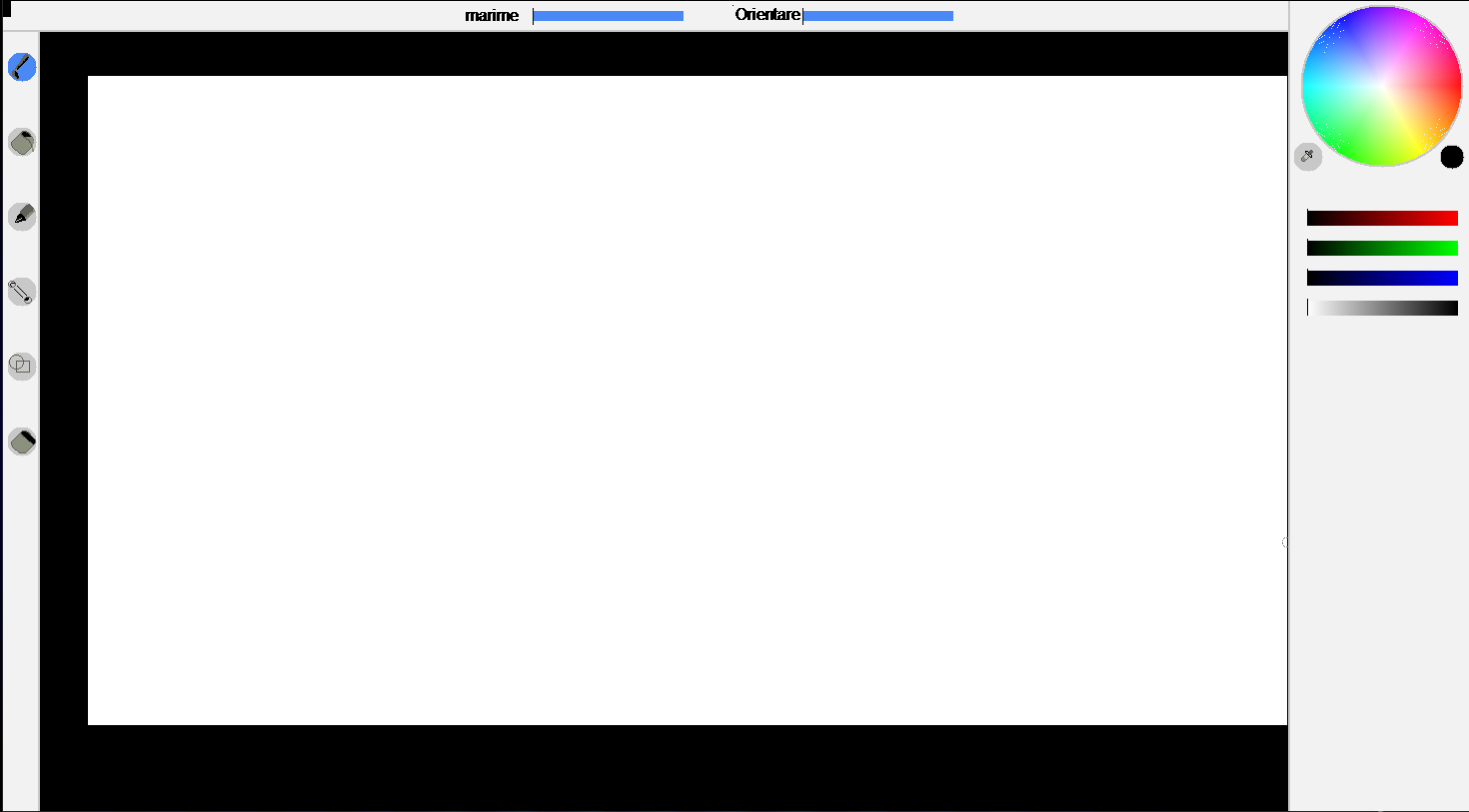
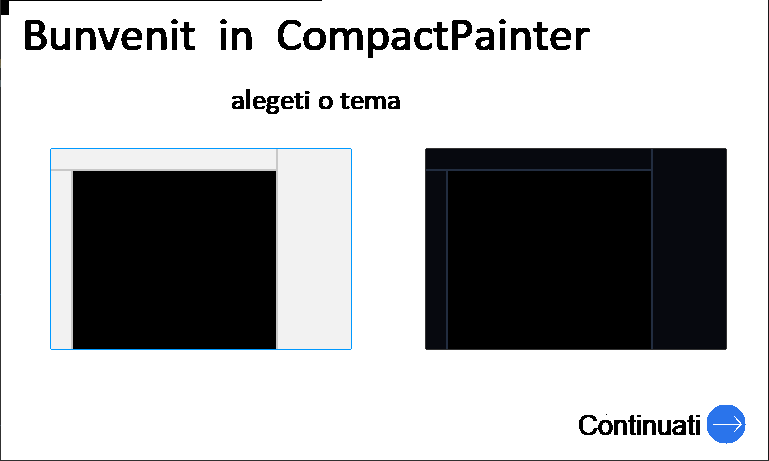
**CompactPainter** este o aplicație în care poți desena cu diferite pensule. Interfața Grafică este user friendly și cât mai simplificată. Pentru a facilita folosirea, există 2 teme pentru interfața grafică, cea nocturnă și cea diurnă.

Ideea programului este de a putea fi creativi într-un mediu foarte portabil, nu este nevoie de instalarea acestuia, mărimea programului nu trece peste 3 MB și posibilitățile pe care programul le poate oferi sunt foarte multe.

Modul prin care utilizatorul interacționează cu programul este prin activarea click-ului drept. Acest program a fost realizat in Microsoft Visual Studio 2019, cu ajutorul următoarelor librării: Iostream, Fstream, Cmath, Thread, Queue,Windows.h.

Pentru convertirea caracterelor în pixeli s-a folosit Microsoft Paint și un program realizat de noi, deși nu este necesar pentru rularea programului principal.

Rapiditatea programului poate fi destul de scazută din cauza funcțiilor GetPixel și SetPixel, care ni s-au părut foarte ok pentru programul acesta și pot fi înțelese de foarte multe persoane fiind ușor de scris.



**Aplicația folosită pentru convertirea caracterelor în pixeli:**

#include <iostream>

#include <fstream>

#include <thread>

#include <windows.h>

using namespace std;

HDC window = GetDC(NULL);

ofstream out("Text.txt");

POINT m,n;

int main() {

GetCursorPos(&m);

while (true) {

if (GetAsyncKeyState((unsigned short)VK\_RBUTTON) & 0x8000) {

cout << 1 << endl;

GetCursorPos(&m);

break;

}

}

this\_thread::sleep\_for(chrono::seconds(1));

while (true) {

if (GetAsyncKeyState((unsigned short)VK\_RBUTTON) & 0x8000) {

cout << 2 << endl;

GetCursorPos(&n);

break;

}

}

long long el=0;

this\_thread::sleep\_for(chrono::seconds(5));

for (int y = m.y; y < n.y; y++) {

printf("%d\n", n.y - y);

for (int x = m.x; x < n.x; x++) {

if (GetPixel(window, x, y) != RGB(255, 255, 255)) {

out << '{' << x - m.x << ',' << y - m.y << "},";

el++;

}

}

}

out << '\n' << el;

out.close();

}

**Partea tehnică a aplicației:**

#include <iostream>

#include <fstream>

#include <cmath>

#include <thread>

#include <queue>

#include <Windows.h>

#define pi 3.14159265358

using namespace std;

HWND consoleWindow = GetConsoleWindow(); // Get a console handle

HDC consoleDC = GetDC(consoleWindow); // Get a handle to device context

RECT rect;

string brush\_t = "paint\_brush";

POINT mouse\_pos;

int switch\_tool = 1;

int th = 1;

double r = 0, g = 0, b = 0;

double shade\_value = 100;

struct brush {

double size = 5;

double orientation = 330;

}paint\_brush, pen, line, eraser;

struct pixel {

double x;

double y;

COLORREF color;

};

int intro[5556][2]{

{5,6},{6,6},{7,6},{8,6},{9,6},{10,6},{11,6},{12,6},{13,6},{14,6},{15,6},{16,6},{17,6},{18,6},{19,6},{129,6},{130,6},{131,6},{132,6},{133,6},{156,6},{157,6},{158,6},{159,6},{160,6},{199,6},{200,6},{201,6},{202,6},{203,6},{204,6},{205,6},{206,6},{207,6},{208,6},{330,6},{331,6},{332,6},{333,6},{334,6},{335,6},{336,6},{337,6},{338,6},{339,6},{340,6},{341,6},{342,6},{343,6},{344,6},{345,6},{370,6},{371,6},{372,6},{373,6},{374,6},{5,7},{6,7},{7,7},{8,7},{9,7},{10,7},{11,7},{12,7},{13,7},{14,7},{15,7},{16,7},{17,7},{18,7},{19,7},{20,7},{129,7},{130,7},{131,7},{132,7},{133,7},{139,7},{140,7},{141,7},{156,7},{157,7},{158,7},{159,7},{160,7},{197,7},{198,7},{199,7},{200,7},{201,7},{202,7},{203,7},{204,7},{205,7},{206,7},{207,7},{208,7},{209,7},{210,7},{322,7},{323,7},{324,7},{330,7},{331,7},{332,7},{333,7},{334,7},{335,7},{336,7},{337,7},{338,7},{339,7},{340,7},{341,7},{342,7},{343,7},{344,7},{345,7},{346,7},{347,7},{370,7},{371,7},{372,7},{373,7},{374,7},{399,7},{400,7},{401,7},{5,8},{6,8},{7,8},{8,8},{9,8},{10,8},{11,8},{12,8},{13,8},{14,8},{15,8},{16,8},{17,8},{18,8},{19,8},{20,8},{21,8},{129,8},{130,8},{131,8},{132,8},{133,8},{137,8},{138,8},{139,8},{140,8},{141,8},{156,8},{157,8},{158,8},{159,8},{160,8},{196,8},{197,8},{198,8},{199,8},{200,8},{201,8},{202,8},{203,8},{204,8},{205,8},{206,8},{207,8},{208,8},{209,8},{210,8},{211,8},{320,8},{321,8},{322,8},{323,8},{324,8},{330,8},{331,8},{332,8},{333,8},{334,8},{335,8},{336,8},{337,8},{338,8},{339,8},{340,8},{341,8},{342,8},{343,8},{344,8},{345,8},{346,8},{347,8},{348,8},{370,8},{371,8},{372,8},{373,8},{374,8},{397,8},{398,8},{399,8},{400,8},{401,8},{5,9},{6,9},{7,9},{8,9},{16,9},{17,9},{18,9},{19,9},{20,9},{21,9},{22,9},{137,9},{138,9},{139,9},{140,9},{141,9},{195,9},{196,9},{197,9},{198,9},{199,9},{200,9},{201,9},{202,9},{206,9},{207,9},{208,9},{209,9},{210,9},{211,9},{212,9},{320,9},{321,9},{322,9},{323,9},{324,9},{330,9},{331,9},{332,9},{333,9},{334,9},{343,9},{344,9},{345,9},{346,9},{347,9},{348,9},{397,9},{398,9},{399,9},{400,9},{401,9},{5,10},{6,10},{7,10},{8,10},{18,10},{19,10},{20,10},{21,10},{22,10},{137,10},{138,10},{139,10},{140,10},{141,10},{194,10},{195,10},{196,10},{197,10},{198,10},{199,10},{208,10},{209,10},{210,10},{211,10},{212,10},{213,10},{320,10},{321,10},{322,10},{323,10},{324,10},{330,10},{331,10},{332,10},{333,10},{334,10},{344,10},{345,10},{346,10},{347,10},{348,10},{349,10},{397,10},{398,10},{399,10},{400,10},{401,10},{5,11},{6,11},{7,11},{8,11},{19,11},{20,11},{21,11},{22,11},{23,11},{137,11},{138,11},{139,11},{140,11},{141,11},{194,11},{195,11},{196,11},{197,11},{198,11},{209,11},{210,11},{211,11},{212,11},{213,11},{320,11},{321,11},{322,11},{323,11},{324,11},{330,11},{331,11},{332,11},{333,11},{334,11},{345,11},{346,11},{347,11},{348,11},{349,11},{397,11},{398,11},{399,11},{400,11},{401,11},{5,12},{6,12},{7,12},{8,12},{19,12},{20,12},{21,12},{22,12},{23,12},{137,12},{138,12},{139,12},{140,12},{141,12},{193,12},{194,12},{195,12},{196,12},{197,12},{210,12},{211,12},{212,12},{213,12},{214,12},{320,12},{321,12},{322,12},{323,12},{324,12},{330,12},{331,12},{332,12},{333,12},{334,12},{345,12},{346,12},{347,12},{348,12},{349,12},{397,12},{398,12},{399,12},{400,12},{401,12},{5,13},{6,13},{7,13},{8,13},{19,13},{20,13},{21,13},{22,13},{28,13},{29,13},{30,13},{31,13},{39,13},{40,13},{41,13},{42,13},{47,13},{48,13},{49,13},{50,13},{52,13},{53,13},{54,13},{55,13},{56,13},{57,13},{58,13},{59,13},{74,13},{75,13},{76,13},{77,13},{87,13},{88,13},{89,13},{90,13},{96,13},{97,13},{98,13},{99,13},{100,13},{101,13},{102,13},{103,13},{111,13},{112,13},{113,13},{114,13},{116,13},{117,13},{118,13},{119,13},{120,13},{121,13},{122,13},{123,13},{129,13},{130,13},{131,13},{132,13},{133,13},{135,13},{136,13},{137,13},{138,13},{139,13},{140,13},{141,13},{142,13},{143,13},{144,13},{156,13},{157,13},{158,13},{159,13},{160,13},{164,13},{165,13},{166,13},{167,13},{169,13},{170,13},{171,13},{172,13},{173,13},{174,13},{175,13},{176,13},{193,13},{194,13},{195,13},{196,13},{197,13},{210,13},{211,13},{212,13},{213,13},{221,13},{222,13},{223,13},{224,13},{225,13},{226,13},{227,13},{228,13},{236,13},{237,13},{238,13},{239,13},{241,13},{242,13},{243,13},{244,13},{245,13},{246,13},{247,13},{252,13},{253,13},{254,13},{255,13},{256,13},{257,13},{258,13},{264,13},{265,13},{266,13},{267,13},{268,13},{270,13},{271,13},{272,13},{273,13},{274,13},{275,13},{276,13},{287,13},{288,13},{289,13},{290,13},{291,13},{292,13},{293,13},{294,13},{295,13},{305,13},{306,13},{307,13},{308,13},{309,13},{310,13},{311,13},{312,13},{318,13},{319,13},{320,13},{321,13},{322,13},{323,13},{324,13},{325,13},{326,13},{327,13},{330,13},{331,13},{332,13},{333,13},{334,13},{345,13},{346,13},{347,13},{348,13},{349,13},{356,13},{357,13},{358,13},{359,13},{360,13},{361,13},{362,13},{363,13},{364,13},{370,13},{371,13},{372,13},{373,13},{374,13},{378,13},{379,13},{380,13},{381,13},{383,13},{384,13},{385,13},{386,13},{387,13},{388,13},{389,13},{390,13},{395,13},{396,13},{397,13},{398,13},{399,13},{400,13},{401,13},{402,13},{403,13},{404,13},{410,13},{411,13},{412,13},{413,13},{414,13},{415,13},{416,13},{417,13},{425,13},{426,13},{427,13},{428,13},{429,13},{430,13},{431,13},{432,13},{433,13},{434,13},{435,13},{5,14},{6,14},{7,14},{8,14},{18,14},{19,14},{20,14},{21,14},{22,14},{28,14},{29,14},{30,14},{31,14},{39,14},{40,14},{41,14},{42,14},{47,14},{48,14},{49,14},{50,14},{51,14},{52,14},{53,14},{54,14},{55,14},{56,14},{57,14},{58,14},{59,14},{60,14},{74,14},{75,14},{76,14},{77,14},{78,14},{86,14},{87,14},{88,14},{89,14},{90,14},{94,14},{95,14},{96,14},{97,14},{98,14},{99,14},{100,14},{101,14},{102,14},{103,14},{104,14},{105,14},{111,14},{112,14},{113,14},{114,14},{115,14},{116,14},{117,14},{118,14},{119,14},{120,14},{121,14},{122,14},{123,14},{124,14},{129,14},{130,14},{131,14},{132,14},{133,14},{135,14},{136,14},{137,14},{138,14},{139,14},{140,14},{141,14},{142,14},{143,14},{144,14},{156,14},{157,14},{158,14},{159,14},{160,14},{164,14},{165,14},{166,14},{167,14},{168,14},{169,14},{170,14},{171,14},{172,14},{173,14},{174,14},{175,14},{176,14},{177,14},{192,14},{193,14},{194,14},{195,14},{196,14},{219,14},{220,14},{221,14},{222,14},{223,14},{224,14},{225,14},{226,14},{227,14},{228,14},{229,14},{230,14},{236,14},{237,14},{238,14},{239,14},{240,14},{241,14},{242,14},{243,14},{244,14},{245,14},{246,14},{247,14},{248,14},{250,14},{251,14},{252,14},{253,14},{254,14},{255,14},{256,14},{257,14},{258,14},{259,14},{264,14},{265,14},{266,14},{267,14},{268,14},{269,14},{270,14},{271,14},{272,14},{273,14},{274,14},{275,14},{276,14},{277,14},{278,14},{285,14},{286,14},{287,14},{288,14},{289,14},{290,14},{291,14},{292,14},{293,14},{294,14},{295,14},{296,14},{304,14},{305,14},{306,14},{307,14},{308,14},{309,14},{310,14},{311,14},{312,14},{313,14},{314,14},{318,14},{319,14},{320,14},{321,14},{322,14},{323,14},{324,14},{325,14},{326,14},{327,14},{330,14},{331,14},{332,14},{333,14},{334,14},{345,14},{346,14},{347,14},{348,14},{349,14},{354,14},{355,14},{356,14},{357,14},{358,14},{359,14},{360,14},{361,14},{362,14},{363,14},{364,14},{365,14},{370,14},{371,14},{372,14},{373,14},{374,14},{378,14},{379,14},{380,14},{381,14},{382,14},{383,14},{384,14},{385,14},{386,14},{387,14},{388,14},{389,14},{390,14},{391,14},{395,14},{396,14},{397,14},{398,14},{399,14},{400,14},{401,14},{402,14},{403,14},{404,14},{408,14},{409,14},{410,14},{411,14},{412,14},{413,14},{414,14},{415,14},{416,14},{417,14},{418,14},{419,14},{425,14},{426,14},{427,14},{428,14},{429,14},{430,14},{431,14},{432,14},{433,14},{434,14},{435,14},{5,15},{6,15},{7,15},{8,15},{16,15},{17,15},{18,15},{19,15},{20,15},{21,15},{28,15},{29,15},{30,15},{31,15},{39,15},{40,15},{41,15},{42,15},{47,15},{48,15},{49,15},{50,15},{51,15},{52,15},{53,15},{54,15},{55,15},{56,15},{57,15},{58,15},{59,15},{60,15},{61,15},{75,15},{76,15},{77,15},{78,15},{86,15},{87,15},{88,15},{89,15},{90,15},{93,15},{94,15},{95,15},{96,15},{97,15},{98,15},{99,15},{100,15},{101,15},{102,15},{103,15},{104,15},{105,15},{111,15},{112,15},{113,15},{114,15},{115,15},{116,15},{117,15},{118,15},{119,15},{120,15},{121,15},{122,15},{123,15},{124,15},{125,15},{129,15},{130,15},{131,15},{132,15},{133,15},{135,15},{136,15},{137,15},{138,15},{139,15},{140,15},{141,15},{142,15},{143,15},{144,15},{156,15},{157,15},{158,15},{159,15},{160,15},{164,15},{165,15},{166,15},{167,15},{168,15},{169,15},{170,15},{171,15},{172,15},{173,15},{174,15},{175,15},{176,15},{177,15},{178,15},{192,15},{193,15},{194,15},{195,15},{196,15},{218,15},{219,15},{220,15},{221,15},{222,15},{223,15},{224,15},{225,15},{226,15},{227,15},{228,15},{229,15},{230,15},{231,15},{236,15},{237,15},{238,15},{239,15},{240,15},{241,15},{242,15},{243,15},{244,15},{245,15},{246,15},{247,15},{248,15},{249,15},{250,15},{251,15},{252,15},{253,15},{254,15},{255,15},{256,15},{257,15},{258,15},{259,15},{260,15},{264,15},{265,15},{266,15},{267,15},{268,15},{269,15},{270,15},{271,15},{272,15},{273,15},{274,15},{275,15},{276,15},{277,15},{278,15},{279,15},{284,15},{285,15},{286,15},{287,15},{288,15},{289,15},{290,15},{291,15},{292,15},{293,15},{294,15},{295,15},{296,15},{297,15},{303,15},{304,15},{305,15},{306,15},{307,15},{308,15},{309,15},{310,15},{311,15},{312,15},{313,15},{314,15},{315,15},{318,15},{319,15},{320,15},{321,15},{322,15},{323,15},{324,15},{325,15},{326,15},{327,15},{330,15},{331,15},{332,15},{333,15},{334,15},{345,15},{346,15},{347,15},{348,15},{349,15},{353,15},{354,15},{355,15},{356,15},{357,15},{358,15},{359,15},{360,15},{361,15},{362,15},{363,15},{364,15},{365,15},{366,15},{370,15},{371,15},{372,15},{373,15},{374,15},{378,15},{379,15},{380,15},{381,15},{382,15},{383,15},{384,15},{385,15},{386,15},{387,15},{388,15},{389,15},{390,15},{391,15},{392,15},{395,15},{396,15},{397,15},{398,15},{399,15},{400,15},{401,15},{402,15},{403,15},{404,15},{407,15},{408,15},{409,15},{410,15},{411,15},{412,15},{413,15},{414,15},{415,15},{416,15},{417,15},{418,15},{419,15},{425,15},{426,15},{427,15},{428,15},{429,15},{430,15},{431,15},{432,15},{433,15},{434,15},{435,15},{5,16},{6,16},{7,16},{8,16},{9,16},{10,16},{11,16},{12,16},{13,16},{14,16},{15,16},{16,16},{17,16},{18,16},{19,16},{20,16},{28,16},{29,16},{30,16},{31,16},{39,16},{40,16},{41,16},{42,16},{47,16},{48,16},{49,16},{50,16},{51,16},{52,16},{53,16},{56,16},{57,16},{58,16},{59,16},{60,16},{61,16},{75,16},{76,16},{77,16},{78,16},{86,16},{87,16},{88,16},{89,16},{93,16},{94,16},{95,16},{96,16},{97,16},{98,16},{101,16},{102,16},{103,16},{104,16},{105,16},{106,16},{111,16},{112,16},{113,16},{114,16},{115,16},{116,16},{117,16},{120,16},{121,16},{122,16},{123,16},{124,16},{125,16},{129,16},{130,16},{131,16},{132,16},{133,16},{137,16},{138,16},{139,16},{140,16},{141,16},{156,16},{157,16},{158,16},{159,16},{160,16},{164,16},{165,16},{166,16},{167,16},{168,16},{169,16},{170,16},{173,16},{174,16},{175,16},{176,16},{177,16},{178,16},{192,16},{193,16},{194,16},{195,16},{196,16},{218,16},{219,16},{220,16},{221,16},{222,16},{223,16},{226,16},{227,16},{228,16},{229,16},{230,16},{231,16},{236,16},{237,16},{238,16},{239,16},{240,16},{241,16},{242,16},{245,16},{246,16},{247,16},{248,16},{249,16},{250,16},{251,16},{252,16},{253,16},{255,16},{256,16},{257,16},{258,16},{259,16},{260,16},{264,16},{265,16},{266,16},{267,16},{268,16},{269,16},{270,16},{271,16},{274,16},{275,16},{276,16},{277,16},{278,16},{279,16},{284,16},{285,16},{286,16},{287,16},{288,16},{289,16},{292,16},{293,16},{294,16},{295,16},{296,16},{297,16},{302,16},{303,16},{304,16},{305,16},{306,16},{307,16},{310,16},{311,16},{312,16},{313,16},{314,16},{315,16},{320,16},{321,16},{322,16},{323,16},{324,16},{330,16},{331,16},{332,16},{333,16},{334,16},{344,16},{345,16},{346,16},{347,16},{348,16},{349,16},{353,16},{354,16},{355,16},{356,16},{357,16},{358,16},{361,16},{362,16},{363,16},{364,16},{365,16},{366,16},{370,16},{371,16},{372,16},{373,16},{374,16},{378,16},{379,16},{380,16},{381,16},{382,16},{383,16},{384,16},{387,16},{388,16},{389,16},{390,16},{391,16},{392,16},{397,16},{398,16},{399,16},{400,16},{401,16},{407,16},{408,16},{409,16},{410,16},{411,16},{412,16},{415,16},{416,16},{417,16},{418,16},{419,16},{420,16},{425,16},{426,16},{427,16},{428,16},{429,16},{430,16},{431,16},{433,16},{434,16},{5,17},{6,17},{7,17},{8,17},{9,17},{10,17},{11,17},{12,17},{13,17},{14,17},{15,17},{16,17},{17,17},{18,17},{19,17},{20,17},{21,17},{28,17},{29,17},{30,17},{31,17},{39,17},{40,17},{41,17},{42,17},{47,17},{48,17},{49,17},{50,17},{51,17},{57,17},{58,17},{59,17},{60,17},{61,17},{75,17},{76,17},{77,17},{78,17},{79,17},{85,17},{86,17},{87,17},{88,17},{89,17},{92,17},{93,17},{94,17},{95,17},{96,17},{103,17},{104,17},{105,17},{106,17},{107,17},{111,17},{112,17},{113,17},{114,17},{115,17},{121,17},{122,17},{123,17},{124,17},{125,17},{129,17},{130,17},{131,17},{132,17},{133,17},{137,17},{138,17},{139,17},{140,17},{141,17},{156,17},{157,17},{158,17},{159,17},{160,17},{164,17},{165,17},{166,17},{167,17},{168,17},{174,17},{175,17},{176,17},{177,17},{178,17},{192,17},{193,17},{194,17},{195,17},{196,17},{217,17},{218,17},{219,17},{220,17},{221,17},{227,17},{228,17},{229,17},{230,17},{231,17},{232,17},{236,17},{237,17},{238,17},{239,17},{240,17},{246,17},{247,17},{248,17},{249,17},{250,17},{251,17},{257,17},{258,17},{259,17},{260,17},{261,17},{264,17},{265,17},{266,17},{267,17},{268,17},{269,17},{270,17},{275,17},{276,17},{277,17},{278,17},{279,17},{280,17},{283,17},{284,17},{285,17},{286,17},{287,17},{294,17},{295,17},{296,17},{297,17},{298,17},{302,17},{303,17},{304,17},{305,17},{306,17},{312,17},{313,17},{314,17},{315,17},{316,17},{320,17},{321,17},{322,17},{323,17},{324,17},{330,17},{331,17},{332,17},{333,17},{334,17},{342,17},{343,17},{344,17},{345,17},{346,17},{347,17},{348,17},{352,17},{353,17},{354,17},{355,17},{356,17},{363,17},{364,17},{365,17},{366,17},{367,17},{370,17},{371,17},{372,17},{373,17},{374,17},{378,17},{379,17},{380,17},{381,17},{382,17},{388,17},{389,17},{390,17},{391,17},{392,17},{397,17},{398,17},{399,17},{400,17},{401,17},{406,17},{407,17},{408,17},{409,17},{410,17},{417,17},{418,17},{419,17},{420,17},{421,17},{425,17},{426,17},{427,17},{428,17},{429,17},{5,18},{6,18},{7,18},{8,18},{9,18},{10,18},{11,18},{12,18},{13,18},{14,18},{15,18},{16,18},{17,18},{18,18},{19,18},{20,18},{21,18},{22,18},{28,18},{29,18},{30,18},{31,18},{39,18},{40,18},{41,18},{42,18},{47,18},{48,18},{49,18},{50,18},{51,18},{58,18},{59,18},{60,18},{61,18},{76,18},{77,18},{78,18},{79,18},{85,18},{86,18},{87,18},{88,18},{92,18},{93,18},{94,18},{95,18},{104,18},{105,18},{106,18},{107,18},{111,18},{112,18},{113,18},{114,18},{115,18},{122,18},{123,18},{124,18},{125,18},{129,18},{130,18},{131,18},{132,18},{133,18},{137,18},{138,18},{139,18},{140,18},{141,18},{156,18},{157,18},{158,18},{159,18},{160,18},{164,18},{165,18},{166,18},{167,18},{168,18},{175,18},{176,18},{177,18},{178,18},{192,18},{193,18},{194,18},{195,18},{196,18},{217,18},{218,18},{219,18},{220,18},{221,18},{228,18},{229,18},{230,18},{231,18},{232,18},{236,18},{237,18},{238,18},{239,18},{240,18},{246,18},{247,18},{248,18},{249,18},{250,18},{257,18},{258,18},{259,18},{260,18},{261,18},{264,18},{265,18},{266,18},{267,18},{268,18},{269,18},{276,18},{277,18},{278,18},{279,18},{280,18},{283,18},{284,18},{285,18},{286,18},{287,18},{294,18},{295,18},{296,18},{297,18},{298,18},{301,18},{302,18},{303,18},{304,18},{305,18},{312,18},{313,18},{314,18},{315,18},{316,18},{320,18},{321,18},{322,18},{323,18},{324,18},{330,18},{331,18},{332,18},{333,18},{334,18},{335,18},{336,18},{337,18},{338,18},{339,18},{340,18},{341,18},{342,18},{343,18},{344,18},{345,18},{346,18},{347,18},{348,18},{352,18},{353,18},{354,18},{355,18},{356,18},{363,18},{364,18},{365,18},{366,18},{367,18},{370,18},{371,18},{372,18},{373,18},{374,18},{378,18},{379,18},{380,18},{381,18},{382,18},{389,18},{390,18},{391,18},{392,18},{397,18},{398,18},{399,18},{400,18},{401,18},{406,18},{407,18},{408,18},{409,18},{418,18},{419,18},{420,18},{421,18},{425,18},{426,18},{427,18},{428,18},{429,18},{5,19},{6,19},{7,19},{8,19},{16,19},{17,19},{18,19},{19,19},{20,19},{21,19},{22,19},{23,19},{28,19},{29,19},{30,19},{31,19},{39,19},{40,19},{41,19},{42,19},{47,19},{48,19},{49,19},{50,19},{58,19},{59,19},{60,19},{61,19},{76,19},{77,19},{78,19},{79,19},{80,19},{84,19},{85,19},{86,19},{87,19},{88,19},{92,19},{93,19},{94,19},{95,19},{104,19},{105,19},{106,19},{107,19},{111,19},{112,19},{113,19},{114,19},{122,19},{123,19},{124,19},{125,19},{129,19},{130,19},{131,19},{132,19},{133,19},{137,19},{138,19},{139,19},{140,19},{141,19},{156,19},{157,19},{158,19},{159,19},{160,19},{164,19},{165,19},{166,19},{167,19},{175,19},{176,19},{177,19},{178,19},{192,19},{193,19},{194,19},{195,19},{196,19},{216,19},{217,19},{218,19},{219,19},{220,19},{229,19},{230,19},{231,19},{232,19},{236,19},{237,19},{238,19},{239,19},{246,19},{247,19},{248,19},{249,19},{250,19},{257,19},{258,19},{259,19},{260,19},{261,19},{264,19},{265,19},{266,19},{267,19},{268,19},{277,19},{278,19},{279,19},{280,19},{294,19},{295,19},{296,19},{297,19},{298,19},{301,19},{302,19},{303,19},{304,19},{305,19},{320,19},{321,19},{322,19},{323,19},{324,19},{330,19},{331,19},{332,19},{333,19},{334,19},{335,19},{336,19},{337,19},{338,19},{339,19},{340,19},{341,19},{342,19},{343,19},{344,19},{345,19},{346,19},{347,19},{363,19},{364,19},{365,19},{366,19},{367,19},{370,19},{371,19},{372,19},{373,19},{374,19},{378,19},{379,19},{380,19},{381,19},{389,19},{390,19},{391,19},{392,19},{397,19},{398,19},{399,19},{400,19},{401,19},{406,19},{407,19},{408,19},{409,19},{418,19},{419,19},{420,19},{421,19},{425,19},{426,19},{427,19},{428,19},{429,19},{5,20},{6,20},{7,20},{8,20},{19,20},{20,20},{21,20},{22,20},{23,20},{28,20},{29,20},{30,20},{31,20},{39,20},{40,20},{41,20},{42,20},{47,20},{48,20},{49,20},{50,20},{58,20},{59,20},{60,20},{61,20},{76,20},{77,20},{78,20},{79,20},{80,20},{84,20},{85,20},{86,20},{87,20},{88,20},{91,20},{92,20},{93,20},{94,20},{95,20},{96,20},{97,20},{98,20},{99,20},{100,20},{101,20},{102,20},{103,20},{104,20},{105,20},{106,20},{107,20},{111,20},{112,20},{113,20},{114,20},{122,20},{123,20},{124,20},{125,20},{129,20},{130,20},{131,20},{132,20},{133,20},{137,20},{138,20},{139,20},{140,20},{141,20},{156,20},{157,20},{158,20},{159,20},{160,20},{164,20},{165,20},{166,20},{167,20},{175,20},{176,20},{177,20},{178,20},{192,20},{193,20},{194,20},{195,20},{196,20},{216,20},{217,20},{218,20},{219,20},{220,20},{229,20},{230,20},{231,20},{232,20},{233,20},{236,20},{237,20},{238,20},{239,20},{246,20},{247,20},{248,20},{249,20},{250,20},{257,20},{258,20},{259,20},{260,20},{261,20},{264,20},{265,20},{266,20},{267,20},{268,20},{277,20},{278,20},{279,20},{280,20},{290,20},{291,20},{292,20},{293,20},{294,20},{295,20},{296,20},{297,20},{298,20},{301,20},{302,20},{303,20},{304,20},{320,20},{321,20},{322,20},{323,20},{324,20},{330,20},{331,20},{332,20},{333,20},{334,20},{335,20},{336,20},{337,20},{338,20},{339,20},{340,20},{341,20},{342,20},{343,20},{344,20},{345,20},{359,20},{360,20},{361,20},{362,20},{363,20},{364,20},{365,20},{366,20},{367,20},{370,20},{371,20},{372,20},{373,20},{374,20},{378,20},{379,20},{380,20},{381,20},{389,20},{390,20},{391,20},{392,20},{397,20},{398,20},{399,20},{400,20},{401,20},{405,20},{406,20},{407,20},{408,20},{409,20},{410,20},{411,20},{412,20},{413,20},{414,20},{415,20},{416,20},{417,20},{418,20},{419,20},{420,20},{421,20},{425,20},{426,20},{427,20},{428,20},{5,21},{6,21},{7,21},{8,21},{20,21},{21,21},{22,21},{23,21},{24,21},{28,21},{29,21},{30,21},{31,21},{39,21},{40,21},{41,21},{42,21},{47,21},{48,21},{49,21},{50,21},{58,21},{59,21},{60,21},{61,21},{77,21},{78,21},{79,21},{80,21},{84,21},{85,21},{86,21},{87,21},{91,21},{92,21},{93,21},{94,21},{95,21},{96,21},{97,21},{98,21},{99,21},{100,21},{101,21},{102,21},{103,21},{104,21},{105,21},{106,21},{107,21},{108,21},{111,21},{112,21},{113,21},{114,21},{122,21},{123,21},{124,21},{125,21},{129,21},{130,21},{131,21},{132,21},{133,21},{137,21},{138,21},{139,21},{140,21},{141,21},{156,21},{157,21},{158,21},{159,21},{160,21},{164,21},{165,21},{166,21},{167,21},{175,21},{176,21},{177,21},{178,21},{192,21},{193,21},{194,21},{195,21},{196,21},{216,21},{217,21},{218,21},{219,21},{220,21},{229,21},{230,21},{231,21},{232,21},{233,21},{236,21},{237,21},{238,21},{239,21},{246,21},{247,21},{248,21},{249,21},{250,21},{257,21},{258,21},{259,21},{260,21},{261,21},{264,21},{265,21},{266,21},{267,21},{268,21},{277,21},{278,21},{279,21},{280,21},{286,21},{287,21},{288,21},{289,21},{290,21},{291,21},{292,21},{293,21},{294,21},{295,21},{296,21},{297,21},{298,21},{301,21},{302,21},{303,21},{304,21},{320,21},{321,21},{322,21},{323,21},{324,21},{330,21},{331,21},{332,21},{333,21},{334,21},{355,21},{356,21},{357,21},{358,21},{359,21},{360,21},{361,21},{362,21},{363,21},{364,21},{365,21},{366,21},{367,21},{370,21},{371,21},{372,21},{373,21},{374,21},{378,21},{379,21},{380,21},{381,21},{389,21},{390,21},{391,21},{392,21},{397,21},{398,21},{399,21},{400,21},{401,21},{405,21},{406,21},{407,21},{408,21},{409,21},{410,21},{411,21},{412,21},{413,21},{414,21},{415,21},{416,21},{417,21},{418,21},{419,21},{420,21},{421,21},{422,21},{425,21},{426,21},{427,21},{428,21},{5,22},{6,22},{7,22},{8,22},{20,22},{21,22},{22,22},{23,22},{24,22},{28,22},{29,22},{30,22},{31,22},{39,22},{40,22},{41,22},{42,22},{47,22},{48,22},{49,22},{50,22},{58,22},{59,22},{60,22},{61,22},{77,22},{78,22},{79,22},{80,22},{81,22},{83,22},{84,22},{85,22},{86,22},{87,22},{91,22},{92,22},{93,22},{94,22},{95,22},{96,22},{97,22},{98,22},{99,22},{100,22},{101,22},{102,22},{103,22},{104,22},{105,22},{106,22},{107,22},{108,22},{111,22},{112,22},{113,22},{114,22},{122,22},{123,22},{124,22},{125,22},{129,22},{130,22},{131,22},{132,22},{133,22},{137,22},{138,22},{139,22},{140,22},{141,22},{156,22},{157,22},{158,22},{159,22},{160,22},{164,22},{165,22},{166,22},{167,22},{175,22},{176,22},{177,22},{178,22},{192,22},{193,22},{194,22},{195,22},{196,22},{211,22},{212,22},{213,22},{214,22},{216,22},{217,22},{218,22},{219,22},{220,22},{229,22},{230,22},{231,22},{232,22},{233,22},{236,22},{237,22},{238,22},{239,22},{246,22},{247,22},{248,22},{249,22},{250,22},{257,22},{258,22},{259,22},{260,22},{261,22},{264,22},{265,22},{266,22},{267,22},{268,22},{277,22},{278,22},{279,22},{280,22},{284,22},{285,22},{286,22},{287,22},{288,22},{289,22},{290,22},{291,22},{292,22},{293,22},{294,22},{295,22},{296,22},{297,22},{298,22},{301,22},{302,22},{303,22},{304,22},{320,22},{321,22},{322,22},{323,22},{324,22},{330,22},{331,22},{332,22},{333,22},{334,22},{353,22},{354,22},{355,22},{356,22},{357,22},{358,22},{359,22},{360,22},{361,22},{362,22},{363,22},{364,22},{365,22},{366,22},{367,22},{370,22},{371,22},{372,22},{373,22},{374,22},{378,22},{379,22},{380,22},{381,22},{389,22},{390,22},{391,22},{392,22},{397,22},{398,22},{399,22},{400,22},{401,22},{405,22},{406,22},{407,22},{408,22},{409,22},{410,22},{411,22},{412,22},{413,22},{414,22},{415,22},{416,22},{417,22},{418,22},{419,22},{420,22},{421,22},{422,22},{425,22},{426,22},{427,22},{428,22},{5,23},{6,23},{7,23},{8,23},{20,23},{21,23},{22,23},{23,23},{24,23},{28,23},{29,23},{30,23},{31,23},{39,23},{40,23},{41,23},{42,23},{47,23},{48,23},{49,23},{50,23},{58,23},{59,23},{60,23},{61,23},{78,23},{79,23},{80,23},{81,23},{83,23},{84,23},{85,23},{86,23},{91,23},{92,23},{93,23},{94,23},{95,23},{111,23},{112,23},{113,23},{114,23},{122,23},{123,23},{124,23},{125,23},{129,23},{130,23},{131,23},{132,23},{133,23},{137,23},{138,23},{139,23},{140,23},{141,23},{156,23},{157,23},{158,23},{159,23},{160,23},{164,23},{165,23},{166,23},{167,23},{175,23},{176,23},{177,23},{178,23},{193,23},{194,23},{195,23},{196,23},{197,23},{210,23},{211,23},{212,23},{213,23},{214,23},{216,23},{217,23},{218,23},{219,23},{220,23},{229,23},{230,23},{231,23},{232,23},{236,23},{237,23},{238,23},{239,23},{246,23},{247,23},{248,23},{249,23},{250,23},{257,23},{258,23},{259,23},{260,23},{261,23},{264,23},{265,23},{266,23},{267,23},{268,23},{277,23},{278,23},{279,23},{280,23},{283,23},{284,23},{285,23},{286,23},{287,23},{288,23},{289,23},{290,23},{291,23},{292,23},{293,23},{294,23},{295,23},{296,23},{297,23},{298,23},{301,23},{302,23},{303,23},{304,23},{320,23},{321,23},{322,23},{323,23},{324,23},{330,23},{331,23},{332,23},{333,23},{334,23},{352,23},{353,23},{354,23},{355,23},{356,23},{357,23},{358,23},{359,23},{360,23},{361,23},{362,23},{363,23},{364,23},{365,23},{366,23},{367,23},{370,23},{371,23},{372,23},{373,23},{374,23},{378,23},{379,23},{380,23},{381,23},{389,23},{390,23},{391,23},{392,23},{397,23},{398,23},{399,23},{400,23},{401,23},{405,23},{406,23},{407,23},{408,23},{409,23},{425,23},{426,23},{427,23},{428,23},{5,24},{6,24},{7,24},{8,24},{20,24},{21,24},{22,24},{23,24},{24,24},{28,24},{29,24},{30,24},{31,24},{39,24},{40,24},{41,24},{42,24},{47,24},{48,24},{49,24},{50,24},{58,24},{59,24},{60,24},{61,24},{78,24},{79,24},{80,24},{81,24},{83,24},{84,24},{85,24},{86,24},{92,24},{93,24},{94,24},{95,24},{111,24},{112,24},{113,24},{114,24},{122,24},{123,24},{124,24},{125,24},{129,24},{130,24},{131,24},{132,24},{133,24},{137,24},{138,24},{139,24},{140,24},{141,24},{156,24},{157,24},{158,24},{159,24},{160,24},{164,24},{165,24},{166,24},{167,24},{175,24},{176,24},{177,24},{178,24},{193,24},{194,24},{195,24},{196,24},{197,24},{210,24},{211,24},{212,24},{213,24},{214,24},{216,24},{217,24},{218,24},{219,24},{220,24},{229,24},{230,24},{231,24},{232,24},{236,24},{237,24},{238,24},{239,24},{246,24},{247,24},{248,24},{249,24},{250,24},{257,24},{258,24},{259,24},{260,24},{261,24},{264,24},{265,24},{266,24},{267,24},{268,24},{277,24},{278,24},{279,24},{280,24},{283,24},{284,24},{285,24},{286,24},{287,24},{288,24},{294,24},{295,24},{296,24},{297,24},{298,24},{301,24},{302,24},{303,24},{304,24},{305,24},{313,24},{314,24},{315,24},{316,24},{320,24},{321,24},{322,24},{323,24},{324,24},{330,24},{331,24},{332,24},{333,24},{334,24},{352,24},{353,24},{354,24},{355,24},{356,24},{357,24},{363,24},{364,24},{365,24},{366,24},{367,24},{370,24},{371,24},{372,24},{373,24},{374,24},{378,24},{379,24},{380,24},{381,24},{389,24},{390,24},{391,24},{392,24},{397,24},{398,24},{399,24},{400,24},{401,24},{406,24},{407,24},{408,24},{409,24},{425,24},{426,24},{427,24},{428,24},{5,25},{6,25},{7,25},{8,25},{19,25},{20,25},{21,25},{22,25},{23,25},{24,25},{28,25},{29,25},{30,25},{31,25},{38,25},{39,25},{40,25},{41,25},{42,25},{47,25},{48,25},{49,25},{50,25},{58,25},{59,25},{60,25},{61,25},{78,25},{79,25},{80,25},{81,25},{82,25},{83,25},{84,25},{85,25},{86,25},{92,25},{93,25},{94,25},{95,25},{96,25},{104,25},{105,25},{106,25},{107,25},{111,25},{112,25},{113,25},{114,25},{122,25},{123,25},{124,25},{125,25},{129,25},{130,25},{131,25},{132,25},{133,25},{137,25},{138,25},{139,25},{140,25},{141,25},{156,25},{157,25},{158,25},{159,25},{160,25},{164,25},{165,25},{166,25},{167,25},{175,25},{176,25},{177,25},{178,25},{194,25},{195,25},{196,25},{197,25},{198,25},{209,25},{210,25},{211,25},{212,25},{213,25},{217,25},{218,25},{219,25},{220,25},{221,25},{228,25},{229,25},{230,25},{231,25},{232,25},{236,25},{237,25},{238,25},{239,25},{246,25},{247,25},{248,25},{249,25},{250,25},{257,25},{258,25},{259,25},{260,25},{261,25},{264,25},{265,25},{266,25},{267,25},{268,25},{269,25},{276,25},{277,25},{278,25},{279,25},{280,25},{283,25},{284,25},{285,25},{286,25},{294,25},{295,25},{296,25},{297,25},{298,25},{301,25},{302,25},{303,25},{304,25},{305,25},{312,25},{313,25},{314,25},{315,25},{316,25},{320,25},{321,25},{322,25},{323,25},{324,25},{330,25},{331,25},{332,25},{333,25},{334,25},{352,25},{353,25},{354,25},{355,25},{363,25},{364,25},{365,25},{366,25},{367,25},{370,25},{371,25},{372,25},{373,25},{374,25},{378,25},{379,25},{380,25},{381,25},{389,25},{390,25},{391,25},{392,25},{397,25},{398,25},{399,25},{400,25},{401,25},{406,25},{407,25},{408,25},{409,25},{410,25},{418,25},{419,25},{420,25},{421,25},{425,25},{426,25},{427,25},{428,25},{5,26},{6,26},{7,26},{8,26},{19,26},{20,26},{21,26},{22,26},{23,26},{28,26},{29,26},{30,26},{31,26},{32,26},{38,26},{39,26},{40,26},{41,26},{42,26},{47,26},{48,26},{49,26},{50,26},{58,26},{59,26},{60,26},{61,26},{79,26},{80,26},{81,26},{82,26},{83,26},{84,26},{85,26},{92,26},{93,26},{94,26},{95,26},{96,26},{103,26},{104,26},{105,26},{106,26},{107,26},{111,26},{112,26},{113,26},{114,26},{122,26},{123,26},{124,26},{125,26},{129,26},{130,26},{131,26},{132,26},{133,26},{137,26},{138,26},{139,26},{140,26},{141,26},{156,26},{157,26},{158,26},{159,26},{160,26},{164,26},{165,26},{166,26},{167,26},{175,26},{176,26},{177,26},{178,26},{194,26},{195,26},{196,26},{197,26},{198,26},{199,26},{208,26},{209,26},{210,26},{211,26},{212,26},{213,26},{217,26},{218,26},{219,26},{220,26},{221,26},{227,26},{228,26},{229,26},{230,26},{231,26},{232,26},{236,26},{237,26},{238,26},{239,26},{246,26},{247,26},{248,26},{249,26},{250,26},{257,26},{258,26},{259,26},{260,26},{261,26},{264,26},{265,26},{266,26},{267,26},{268,26},{269,26},{275,26},{276,26},{277,26},{278,26},{279,26},{280,26},{282,26},{283,26},{284,26},{285,26},{286,26},{293,26},{294,26},{295,26},{296,26},{297,26},{298,26},{302,26},{303,26},{304,26},{305,26},{306,26},{312,26},{313,26},{314,26},{315,26},{316,26},{320,26},{321,26},{322,26},{323,26},{324,26},{330,26},{331,26},{332,26},{333,26},{334,26},{351,26},{352,26},{353,26},{354,26},{355,26},{362,26},{363,26},{364,26},{365,26},{366,26},{367,26},{370,26},{371,26},{372,26},{373,26},{374,26},{378,26},{379,26},{380,26},{381,26},{389,26},{390,26},{391,26},{392,26},{397,26},{398,26},{399,26},{400,26},{401,26},{406,26},{407,26},{408,26},{409,26},{410,26},{417,26},{418,26},{419,26},{420,26},{421,26},{425,26},{426,26},{427,26},{428,26},{5,27},{6,27},{7,27},{8,27},{17,27},{18,27},{19,27},{20,27},{21,27},{22,27},{23,27},{28,27},{29,27},{30,27},{31,27},{32,27},{33,27},{36,27},{37,27},{38,27},{39,27},{40,27},{41,27},{42,27},{47,27},{48,27},{49,27},{50,27},{58,27},{59,27},{60,27},{61,27},{79,27},{80,27},{81,27},{82,27},{83,27},{84,27},{85,27},{93,27},{94,27},{95,27},{96,27},{97,27},{98,27},{101,27},{102,27},{103,27},{104,27},{105,27},{106,27},{111,27},{112,27},{113,27},{114,27},{122,27},{123,27},{124,27},{125,27},{129,27},{130,27},{131,27},{132,27},{133,27},{137,27},{138,27},{139,27},{140,27},{141,27},{156,27},{157,27},{158,27},{159,27},{160,27},{164,27},{165,27},{166,27},{167,27},{175,27},{176,27},{177,27},{178,27},{195,27},{196,27},{197,27},{198,27},{199,27},{200,27},{201,27},{202,27},{206,27},{207,27},{208,27},{209,27},{210,27},{211,27},{212,27},{218,27},{219,27},{220,27},{221,27},{222,27},{223,27},{226,27},{227,27},{228,27},{229,27},{230,27},{231,27},{236,27},{237,27},{238,27},{239,27},{246,27},{247,27},{248,27},{249,27},{250,27},{257,27},{258,27},{259,27},{260,27},{261,27},{264,27},{265,27},{266,27},{267,27},{268,27},{269,27},{270,27},{271,27},{274,27},{275,27},{276,27},{277,27},{278,27},{279,27},{283,27},{284,27},{285,27},{286,27},{287,27},{288,27},{291,27},{292,27},{293,27},{294,27},{295,27},{296,27},{297,27},{298,27},{302,27},{303,27},{304,27},{305,27},{306,27},{307,27},{310,27},{311,27},{312,27},{313,27},{314,27},{315,27},{320,27},{321,27},{322,27},{323,27},{324,27},{330,27},{331,27},{332,27},{333,27},{334,27},{352,27},{353,27},{354,27},{355,27},{356,27},{357,27},{360,27},{361,27},{362,27},{363,27},{364,27},{365,27},{366,27},{367,27},{370,27},{371,27},{372,27},{373,27},{374,27},{378,27},{379,27},{380,27},{381,27},{389,27},{390,27},{391,27},{392,27},{397,27},{398,27},{399,27},{400,27},{401,27},{407,27},{408,27},{409,27},{410,27},{411,27},{412,27},{415,27},{416,27},{417,27},{418,27},{419,27},{420,27},{425,27},{426,27},{427,27},{428,27},{5,28},{6,28},{7,28},{8,28},{9,28},{10,28},{11,28},{12,28},{13,28},{14,28},{15,28},{16,28},{17,28},{18,28},{19,28},{20,28},{21,28},{22,28},{28,28},{29,28},{30,28},{31,28},{32,28},{33,28},{34,28},{35,28},{36,28},{37,28},{38,28},{39,28},{40,28},{41,28},{42,28},{47,28},{48,28},{49,28},{50,28},{58,28},{59,28},{60,28},{61,28},{80,28},{81,28},{82,28},{83,28},{84,28},{85,28},{93,28},{94,28},{95,28},{96,28},{97,28},{98,28},{99,28},{100,28},{101,28},{102,28},{103,28},{104,28},{105,28},{106,28},{111,28},{112,28},{113,28},{114,28},{122,28},{123,28},{124,28},{125,28},{129,28},{130,28},{131,28},{132,28},{133,28},{137,28},{138,28},{139,28},{140,28},{141,28},{142,28},{143,28},{144,28},{156,28},{157,28},{158,28},{159,28},{160,28},{164,28},{165,28},{166,28},{167,28},{175,28},{176,28},{177,28},{178,28},{196,28},{197,28},{198,28},{199,28},{200,28},{201,28},{202,28},{203,28},{204,28},{205,28},{206,28},{207,28},{208,28},{209,28},{210,28},{211,28},{218,28},{219,28},{220,28},{221,28},{222,28},{223,28},{224,28},{225,28},{226,28},{227,28},{228,28},{229,28},{230,28},{231,28},{236,28},{237,28},{238,28},{239,28},{246,28},{247,28},{248,28},{249,28},{250,28},{257,28},{258,28},{259,28},{260,28},{261,28},{264,28},{265,28},{266,28},{267,28},{268,28},{269,28},{270,28},{271,28},{272,28},{273,28},{274,28},{275,28},{276,28},{277,28},{278,28},{283,28},{284,28},{285,28},{286,28},{287,28},{288,28},{289,28},{290,28},{291,28},{292,28},{293,28},{294,28},{295,28},{296,28},{297,28},{298,28},{303,28},{304,28},{305,28},{306,28},{307,28},{308,28},{309,28},{310,28},{311,28},{312,28},{313,28},{314,28},{315,28},{320,28},{321,28},{322,28},{323,28},{324,28},{325,28},{326,28},{327,28},{330,28},{331,28},{332,28},{333,28},{334,28},{352,28},{353,28},{354,28},{355,28},{356,28},{357,28},{358,28},{359,28},{360,28},{361,28},{362,28},{363,28},{364,28},{365,28},{366,28},{367,28},{370,28},{371,28},{372,28},{373,28},{374,28},{378,28},{379,28},{380,28},{381,28},{389,28},{390,28},{391,28},{392,28},{397,28},{398,28},{399,28},{400,28},{401,28},{402,28},{403,28},{404,28},{407,28},{408,28},{409,28},{410,28},{411,28},{412,28},{413,28},{414,28},{415,28},{416,28},{417,28},{418,28},{419,28},{420,28},{425,28},{426,28},{427,28},{428,28},{5,29},{6,29},{7,29},{8,29},{9,29},{10,29},{11,29},{12,29},{13,29},{14,29},{15,29},{16,29},{17,29},{18,29},{19,29},{20,29},{21,29},{29,29},{30,29},{31,29},{32,29},{33,29},{34,29},{35,29},{36,29},{37,29},{38,29},{39,29},{40,29},{41,29},{42,29},{47,29},{48,29},{49,29},{50,29},{58,29},{59,29},{60,29},{61,29},{80,29},{81,29},{82,29},{83,29},{84,29},{94,29},{95,29},{96,29},{97,29},{98,29},{99,29},{100,29},{101,29},{102,29},{103,29},{104,29},{105,29},{111,29},{112,29},{113,29},{114,29},{122,29},{123,29},{124,29},{125,29},{129,29},{130,29},{131,29},{132,29},{133,29},{138,29},{139,29},{140,29},{141,29},{142,29},{143,29},{144,29},{156,29},{157,29},{158,29},{159,29},{160,29},{164,29},{165,29},{166,29},{167,29},{175,29},{176,29},{177,29},{178,29},{197,29},{198,29},{199,29},{200,29},{201,29},{202,29},{203,29},{204,29},{205,29},{206,29},{207,29},{208,29},{209,29},{210,29},{219,29},{220,29},{221,29},{222,29},{223,29},{224,29},{225,29},{226,29},{227,29},{228,29},{229,29},{230,29},{236,29},{237,29},{238,29},{239,29},{246,29},{247,29},{248,29},{249,29},{250,29},{257,29},{258,29},{259,29},{260,29},{261,29},{264,29},{265,29},{266,29},{267,29},{268,29},{269,29},{270,29},{271,29},{272,29},{273,29},{274,29},{275,29},{276,29},{277,29},{284,29},{285,29},{286,29},{287,29},{288,29},{289,29},{290,29},{291,29},{292,29},{293,29},{294,29},{295,29},{296,29},{297,29},{298,29},{304,29},{305,29},{306,29},{307,29},{308,29},{309,29},{310,29},{311,29},{312,29},{313,29},{314,29},{321,29},{322,29},{323,29},{324,29},{325,29},{326,29},{327,29},{330,29},{331,29},{332,29},{333,29},{334,29},{353,29},{354,29},{355,29},{356,29},{357,29},{358,29},{359,29},{360,29},{361,29},{362,29},{363,29},{364,29},{365,29},{366,29},{367,29},{370,29},{371,29},{372,29},{373,29},{374,29},{378,29},{379,29},{380,29},{381,29},{389,29},{390,29},{391,29},{392,29},{398,29},{399,29},{400,29},{401,29},{402,29},{403,29},{404,29},{408,29},{409,29},{410,29},{411,29},{412,29},{413,29},{414,29},{415,29},{416,29},{417,29},{418,29},{419,29},{425,29},{426,29},{427,29},{428,29},{5,30},{6,30},{7,30},{8,30},{9,30},{10,30},{11,30},{12,30},{13,30},{14,30},{15,30},{16,30},{17,30},{18,30},{19,30},{30,30},{31,30},{32,30},{33,30},{34,30},{35,30},{36,30},{37,30},{39,30},{40,30},{41,30},{42,30},{47,30},{48,30},{49,30},{50,30},{58,30},{59,30},{60,30},{61,30},{80,30},{81,30},{82,30},{83,30},{84,30},{96,30},{97,30},{98,30},{99,30},{100,30},{101,30},{102,30},{103,30},{111,30},{112,30},{113,30},{114,30},{122,30},{123,30},{124,30},{125,30},{129,30},{130,30},{131,30},{132,30},{133,30},{139,30},{140,30},{141,30},{142,30},{143,30},{144,30},{156,30},{157,30},{158,30},{159,30},{160,30},{164,30},{165,30},{166,30},{167,30},{175,30},{176,30},{177,30},{178,30},{199,30},{200,30},{201,30},{202,30},{203,30},{204,30},{205,30},{206,30},{207,30},{208,30},{221,30},{222,30},{223,30},{224,30},{225,30},{226,30},{227,30},{228,30},{236,30},{237,30},{238,30},{239,30},{246,30},{247,30},{248,30},{249,30},{250,30},{257,30},{258,30},{259,30},{260,30},{261,30},{264,30},{265,30},{266,30},{267,30},{268,30},{270,30},{271,30},{272,30},{273,30},{274,30},{275,30},{276,30},{285,30},{286,30},{287,30},{288,30},{289,30},{290,30},{291,30},{292,30},{293,30},{295,30},{296,30},{297,30},{298,30},{305,30},{306,30},{307,30},{308,30},{309,30},{310,30},{311,30},{312,30},{322,30},{323,30},{324,30},{325,30},{326,30},{327,30},{330,30},{331,30},{332,30},{333,30},{334,30},{354,30},{355,30},{356,30},{357,30},{358,30},{359,30},{360,30},{361,30},{362,30},{364,30},{365,30},{366,30},{367,30},{370,30},{371,30},{372,30},{373,30},{374,30},{378,30},{379,30},{380,30},{381,30},{389,30},{390,30},{391,30},{392,30},{399,30},{400,30},{401,30},{402,30},{403,30},{404,30},{410,30},{411,30},{412,30},{413,30},{414,30},{415,30},{416,30},{417,30},{425,30},{426,30},{427,30},{428,30},{264,31},{265,31},{266,31},{267,31},{268,31},{264,32},{265,32},{266,32},{267,32},{268,32},{264,33},{265,33},{266,33},{267,33},{268,33},{264,34},{265,34},{266,34},{267,34},{268,34},{264,35},{265,35},{266,35},{267,35},{268,35},{264,36},{265,36},{266,36},{267,36},{268,36},{264,37},{265,37},{266,37},{267,37},{268,37},{185,53},{186,53},{224,53},{225,53},{137,54},{138,54},{139,54},{140,54},{184,54},{185,54},{186,54},{190,54},{191,54},{192,54},{223,54},{224,54},{225,54},{137,55},{138,55},{139,55},{140,55},{183,55},{184,55},{185,55},{186,55},{190,55},{191,55},{192,55},{222,55},{223,55},{224,55},{225,55},{137,56},{138,56},{139,56},{140,56},{183,56},{184,56},{185,56},{186,56},{222,56},{223,56},{224,56},{225,56},{137,57},{138,57},{139,57},{140,57},{183,57},{184,57},{185,57},{186,57},{222,57},{223,57},{224,57},{225,57},{126,58},{127,58},{128,58},{129,58},{130,58},{131,58},{132,58},{133,58},{137,58},{138,58},{139,58},{140,58},{145,58},{146,58},{147,58},{148,58},{149,58},{150,58},{151,58},{158,58},{159,58},{160,58},{161,58},{162,58},{163,58},{164,58},{165,58},{166,58},{167,58},{172,58},{173,58},{174,58},{175,58},{176,58},{177,58},{178,58},{182,58},{183,58},{184,58},{185,58},{186,58},{187,58},{188,58},{190,58},{191,58},{192,58},{204,58},{205,58},{206,58},{207,58},{208,58},{209,58},{210,58},{221,58},{222,58},{223,58},{224,58},{225,58},{226,58},{227,58},{231,58},{232,58},{233,58},{234,58},{235,58},{236,58},{237,58},{242,58},{243,58},{244,58},{245,58},{246,58},{247,58},{248,58},{249,58},{250,58},{253,58},{254,58},{255,58},{256,58},{257,58},{258,58},{264,58},{265,58},{266,58},{267,58},{268,58},{269,58},{270,58},{271,58},{125,59},{126,59},{127,59},{128,59},{129,59},{130,59},{131,59},{132,59},{133,59},{134,59},{137,59},{138,59},{139,59},{140,59},{144,59},{145,59},{146,59},{147,59},{148,59},{149,59},{150,59},{151,59},{152,59},{157,59},{158,59},{159,59},{160,59},{161,59},{162,59},{163,59},{164,59},{165,59},{166,59},{167,59},{171,59},{172,59},{173,59},{174,59},{175,59},{176,59},{177,59},{178,59},{179,59},{182,59},{183,59},{184,59},{185,59},{186,59},{187,59},{188,59},{190,59},{191,59},{192,59},{203,59},{204,59},{205,59},{206,59},{207,59},{208,59},{209,59},{210,59},{211,59},{221,59},{222,59},{223,59},{224,59},{225,59},{226,59},{227,59},{230,59},{231,59},{232,59},{233,59},{234,59},{235,59},{236,59},{237,59},{238,59},{242,59},{243,59},{244,59},{245,59},{246,59},{247,59},{248,59},{249,59},{250,59},{251,59},{252,59},{253,59},{254,59},{255,59},{256,59},{257,59},{258,59},{259,59},{263,59},{264,59},{265,59},{266,59},{267,59},{268,59},{269,59},{270,59},{271,59},{272,59},{124,60},{125,60},{126,60},{127,60},{130,60},{131,60},{132,60},{133,60},{134,60},{137,60},{138,60},{139,60},{140,60},{143,60},{144,60},{145,60},{146,60},{147,60},{149,60},{150,60},{151,60},{152,60},{153,60},{156,60},{157,60},{158,60},{159,60},{160,60},{162,60},{163,60},{164,60},{165,60},{166,60},{167,60},{170,60},{171,60},{172,60},{173,60},{174,60},{176,60},{177,60},{178,60},{179,60},{180,60},{183,60},{184,60},{185,60},{186,60},{190,60},{191,60},{192,60},{202,60},{203,60},{204,60},{205,60},{206,60},{208,60},{209,60},{210,60},{211,60},{212,60},{222,60},{223,60},{224,60},{225,60},{229,60},{230,60},{231,60},{232,60},{233,60},{235,60},{236,60},{237,60},{238,60},{239,60},{242,60},{243,60},{244,60},{245,60},{246,60},{248,60},{249,60},{250,60},{251,60},{252,60},{253,60},{256,60},{257,60},{258,60},{259,60},{262,60},{263,60},{264,60},{265,60},{268,60},{269,60},{270,60},{271,60},{272,60},{124,61},{125,61},{126,61},{131,61},{132,61},{133,61},{134,61},{137,61},{138,61},{139,61},{140,61},{143,61},{144,61},{145,61},{151,61},{152,61},{153,61},{156,61},{157,61},{158,61},{159,61},{164,61},{165,61},{166,61},{167,61},{170,61},{171,61},{172,61},{178,61},{179,61},{180,61},{183,61},{184,61},{185,61},{186,61},{190,61},{191,61},{192,61},{202,61},{203,61},{204,61},{205,61},{210,61},{211,61},{212,61},{222,61},{223,61},{224,61},{225,61},{229,61},{230,61},{231,61},{237,61},{238,61},{239,61},{242,61},{243,61},{244,61},{245,61},{249,61},{250,61},{251,61},{252,61},{256,61},{257,61},{258,61},{259,61},{262,61},{263,61},{264,61},{269,61},{270,61},{271,61},{272,61},{131,62},{132,62},{133,62},{134,62},{137,62},{138,62},{139,62},{140,62},{142,62},{143,62},{144,62},{145,62},{151,62},{152,62},{153,62},{154,62},{156,62},{157,62},{158,62},{164,62},{165,62},{166,62},{167,62},{169,62},{170,62},{171,62},{172,62},{178,62},{179,62},{180,62},{181,62},{183,62},{184,62},{185,62},{186,62},{190,62},{191,62},{192,62},{201,62},{202,62},{203,62},{204,62},{210,62},{211,62},{212,62},{213,62},{222,62},{223,62},{224,62},{225,62},{228,62},{229,62},{230,62},{231,62},{237,62},{238,62},{239,62},{240,62},{242,62},{243,62},{244,62},{245,62},{249,62},{250,62},{251,62},{252,62},{257,62},{258,62},{259,62},{269,62},{270,62},{271,62},{272,62},{127,63},{128,63},{129,63},{130,63},{131,63},{132,63},{133,63},{134,63},{137,63},{138,63},{139,63},{140,63},{142,63},{143,63},{144,63},{145,63},{146,63},{147,63},{148,63},{149,63},{150,63},{151,63},{152,63},{153,63},{154,63},{155,63},{156,63},{157,63},{158,63},{165,63},{166,63},{167,63},{169,63},{170,63},{171,63},{172,63},{173,63},{174,63},{175,63},{176,63},{177,63},{178,63},{179,63},{180,63},{181,63},{183,63},{184,63},{185,63},{186,63},{190,63},{191,63},{192,63},{201,63},{202,63},{203,63},{204,63},{210,63},{211,63},{212,63},{213,63},{222,63},{223,63},{224,63},{225,63},{228,63},{229,63},{230,63},{231,63},{232,63},{233,63},{234,63},{235,63},{236,63},{237,63},{238,63},{239,63},{240,63},{242,63},{243,63},{244,63},{249,63},{250,63},{251,63},{252,63},{257,63},{258,63},{259,63},{265,63},{266,63},{267,63},{268,63},{269,63},{270,63},{271,63},{272,63},{125,64},{126,64},{127,64},{128,64},{129,64},{130,64},{131,64},{132,64},{133,64},{134,64},{137,64},{138,64},{139,64},{140,64},{142,64},{143,64},{144,64},{145,64},{146,64},{147,64},{148,64},{149,64},{150,64},{151,64},{152,64},{153,64},{154,64},{155,64},{156,64},{157,64},{158,64},{165,64},{166,64},{167,64},{169,64},{170,64},{171,64},{172,64},{173,64},{174,64},{175,64},{176,64},{177,64},{178,64},{179,64},{180,64},{181,64},{183,64},{184,64},{185,64},{186,64},{190,64},{191,64},{192,64},{201,64},{202,64},{203,64},{204,64},{210,64},{211,64},{212,64},{213,64},{222,64},{223,64},{224,64},{225,64},{228,64},{229,64},{230,64},{231,64},{232,64},{233,64},{234,64},{235,64},{236,64},{237,64},{238,64},{239,64},{240,64},{242,64},{243,64},{244,64},{249,64},{250,64},{251,64},{252,64},{257,64},{258,64},{259,64},{263,64},{264,64},{265,64},{266,64},{267,64},{268,64},{269,64},{270,64},{271,64},{272,64},{124,65},{125,65},{126,65},{127,65},{128,65},{129,65},{130,65},{131,65},{132,65},{133,65},{134,65},{137,65},{138,65},{139,65},{140,65},{142,65},{143,65},{144,65},{145,65},{155,65},{156,65},{157,65},{158,65},{165,65},{166,65},{167,65},{169,65},{170,65},{171,65},{172,65},{183,65},{184,65},{185,65},{186,65},{190,65},{191,65},{192,65},{201,65},{202,65},{203,65},{204,65},{210,65},{211,65},{212,65},{213,65},{222,65},{223,65},{224,65},{225,65},{228,65},{229,65},{230,65},{231,65},{242,65},{243,65},{244,65},{249,65},{250,65},{251,65},{252,65},{257,65},{258,65},{259,65},{262,65},{263,65},{264,65},{265,65},{266,65},{267,65},{268,65},{269,65},{270,65},{271,65},{272,65},{123,66},{124,66},{125,66},{126,66},{131,66},{132,66},{133,66},{134,66},{137,66},{138,66},{139,66},{140,66},{142,66},{143,66},{144,66},{145,66},{156,66},{157,66},{158,66},{164,66},{165,66},{166,66},{167,66},{169,66},{170,66},{171,66},{172,66},{183,66},{184,66},{185,66},{186,66},{190,66},{191,66},{192,66},{201,66},{202,66},{203,66},{204,66},{210,66},{211,66},{212,66},{213,66},{222,66},{223,66},{224,66},{225,66},{228,66},{229,66},{230,66},{231,66},{242,66},{243,66},{244,66},{249,66},{250,66},{251,66},{252,66},{257,66},{258,66},{259,66},{261,66},{262,66},{263,66},{264,66},{269,66},{270,66},{271,66},{272,66},{123,67},{124,67},{125,67},{126,67},{131,67},{132,67},{133,67},{134,67},{137,67},{138,67},{139,67},{140,67},{143,67},{144,67},{145,67},{146,67},{151,67},{152,67},{153,67},{154,67},{156,67},{157,67},{158,67},{159,67},{164,67},{165,67},{166,67},{167,67},{170,67},{171,67},{172,67},{173,67},{178,67},{179,67},{180,67},{181,67},{183,67},{184,67},{185,67},{186,67},{190,67},{191,67},{192,67},{202,67},{203,67},{204,67},{205,67},{209,67},{210,67},{211,67},{212,67},{222,67},{223,67},{224,67},{225,67},{229,67},{230,67},{231,67},{232,67},{237,67},{238,67},{239,67},{240,67},{242,67},{243,67},{244,67},{249,67},{250,67},{251,67},{252,67},{257,67},{258,67},{259,67},{261,67},{262,67},{263,67},{264,67},{269,67},{270,67},{271,67},{272,67},{123,68},{124,68},{125,68},{126,68},{127,68},{129,68},{130,68},{131,68},{132,68},{133,68},{134,68},{137,68},{138,68},{139,68},{140,68},{143,68},{144,68},{145,68},{146,68},{147,68},{150,68},{151,68},{152,68},{153,68},{156,68},{157,68},{158,68},{159,68},{160,68},{162,68},{163,68},{164,68},{165,68},{166,68},{167,68},{170,68},{171,68},{172,68},{173,68},{174,68},{177,68},{178,68},{179,68},{180,68},{183,68},{184,68},{185,68},{186,68},{190,68},{191,68},{192,68},{202,68},{203,68},{204,68},{205,68},{206,68},{208,68},{209,68},{210,68},{211,68},{212,68},{222,68},{223,68},{224,68},{225,68},{229,68},{230,68},{231,68},{232,68},{233,68},{236,68},{237,68},{238,68},{239,68},{242,68},{243,68},{244,68},{249,68},{250,68},{251,68},{252,68},{257,68},{258,68},{259,68},{261,68},{262,68},{263,68},{264,68},{265,68},{267,68},{268,68},{269,68},{270,68},{271,68},{272,68},{124,69},{125,69},{126,69},{127,69},{128,69},{129,69},{130,69},{131,69},{132,69},{133,69},{134,69},{137,69},{138,69},{139,69},{140,69},{144,69},{145,69},{146,69},{147,69},{148,69},{149,69},{150,69},{151,69},{152,69},{157,69},{158,69},{159,69},{160,69},{161,69},{162,69},{163,69},{164,69},{165,69},{166,69},{167,69},{171,69},{172,69},{173,69},{174,69},{175,69},{176,69},{177,69},{178,69},{179,69},{184,69},{185,69},{186,69},{187,69},{188,69},{190,69},{191,69},{192,69},{203,69},{204,69},{205,69},{206,69},{207,69},{208,69},{209,69},{210,69},{211,69},{223,69},{224,69},{225,69},{226,69},{227,69},{230,69},{231,69},{232,69},{233,69},{234,69},{235,69},{236,69},{237,69},{238,69},{242,69},{243,69},{244,69},{249,69},{250,69},{251,69},{252,69},{257,69},{258,69},{259,69},{262,69},{263,69},{264,69},{265,69},{266,69},{267,69},{268,69},{269,69},{270,69},{271,69},{272,69},{125,70},{126,70},{127,70},{128,70},{129,70},{130,70},{132,70},{133,70},{134,70},{135,70},{137,70},{138,70},{139,70},{140,70},{145,70},{146,70},{147,70},{148,70},{149,70},{150,70},{151,70},{158,70},{159,70},{160,70},{161,70},{162,70},{163,70},{164,70},{165,70},{166,70},{167,70},{172,70},{173,70},{174,70},{175,70},{176,70},{177,70},{178,70},{184,70},{185,70},{186,70},{187,70},{188,70},{190,70},{191,70},{192,70},{204,70},{205,70},{206,70},{207,70},{208,70},{209,70},{210,70},{223,70},{224,70},{225,70},{226,70},{227,70},{231,70},{232,70},{233,70},{234,70},{235,70},{236,70},{237,70},{242,70},{243,70},{244,70},{249,70},{250,70},{251,70},{252,70},{257,70},{258,70},{259,70},{263,70},{264,70},{265,70},{266,70},{267,70},{268,70},{270,70},{271,70},{272,70},{273,70},{164,71},{165,71},{166,71},{167,71},{156,72},{157,72},{158,72},{164,72},{165,72},{166,72},{167,72},{156,73},{157,73},{158,73},{159,73},{160,73},{163,73},{164,73},{165,73},{166,73},{156,74},{157,74},{158,74},{159,74},{160,74},{161,74},{162,74},{163,74},{164,74},{165,74},{166,74},{158,75},{159,75},{160,75},{161,75},{162,75},{163,75},{164,75},{165,75},

};

int continuati[1134][2]{

{8,1},{9,1},{10,1},{11,1},{12,1},{13,1},{14,1},{15,1},{16,1},{55,1},{56,1},{57,1},{61,1},{62,1},{63,1},{64,1},{114,1},{115,1},{116,1},{120,1},{121,1},{122,1},{123,1},{6,2},{7,2},{8,2},{9,2},{10,2},{11,2},{12,2},{13,2},{14,2},{15,2},{16,2},{17,2},{18,2},{54,2},{55,2},{56,2},{57,2},{61,2},{62,2},{63,2},{64,2},{113,2},{114,2},{115,2},{116,2},{120,2},{121,2},{122,2},{123,2},{5,3},{6,3},{7,3},{8,3},{9,3},{10,3},{14,3},{15,3},{16,3},{17,3},{18,3},{19,3},{54,3},{55,3},{56,3},{57,3},{113,3},{114,3},{115,3},{116,3},{5,4},{6,4},{7,4},{8,4},{16,4},{17,4},{18,4},{19,4},{54,4},{55,4},{56,4},{57,4},{113,4},{114,4},{115,4},{116,4},{4,5},{5,5},{6,5},{7,5},{17,5},{18,5},{19,5},{20,5},{54,5},{55,5},{56,5},{57,5},{113,5},{114,5},{115,5},{116,5},{3,6},{4,6},{5,6},{6,6},{17,6},{18,6},{19,6},{20,6},{26,6},{27,6},{28,6},{29,6},{30,6},{31,6},{32,6},{38,6},{39,6},{40,6},{41,6},{42,6},{43,6},{44,6},{45,6},{46,6},{47,6},{48,6},{52,6},{53,6},{54,6},{55,6},{56,6},{57,6},{58,6},{59,6},{61,6},{62,6},{63,6},{64,6},{67,6},{68,6},{69,6},{70,6},{71,6},{72,6},{73,6},{74,6},{75,6},{76,6},{77,6},{82,6},{83,6},{84,6},{85,6},{91,6},{92,6},{93,6},{94,6},{99,6},{100,6},{101,6},{102,6},{103,6},{104,6},{105,6},{106,6},{107,6},{111,6},{112,6},{113,6},{114,6},{115,6},{116,6},{117,6},{118,6},{120,6},{121,6},{122,6},{123,6},{3,7},{4,7},{5,7},{6,7},{24,7},{25,7},{26,7},{27,7},{28,7},{29,7},{30,7},{31,7},{32,7},{33,7},{34,7},{38,7},{39,7},{40,7},{41,7},{42,7},{43,7},{44,7},{45,7},{46,7},{47,7},{48,7},{49,7},{52,7},{53,7},{54,7},{55,7},{56,7},{57,7},{58,7},{59,7},{61,7},{62,7},{63,7},{64,7},{67,7},{68,7},{69,7},{70,7},{71,7},{72,7},{73,7},{74,7},{75,7},{76,7},{77,7},{78,7},{82,7},{83,7},{84,7},{85,7},{91,7},{92,7},{93,7},{94,7},{98,7},{99,7},{100,7},{101,7},{102,7},{103,7},{104,7},{105,7},{106,7},{107,7},{108,7},{111,7},{112,7},{113,7},{114,7},{115,7},{116,7},{117,7},{118,7},{120,7},{121,7},{122,7},{123,7},{3,8},{4,8},{5,8},{6,8},{24,8},{25,8},{26,8},{27,8},{28,8},{30,8},{31,8},{32,8},{33,8},{34,8},{38,8},{39,8},{40,8},{41,8},{42,8},{43,8},{45,8},{46,8},{47,8},{48,8},{49,8},{54,8},{55,8},{56,8},{57,8},{61,8},{62,8},{63,8},{64,8},{67,8},{68,8},{69,8},{70,8},{71,8},{72,8},{74,8},{75,8},{76,8},{77,8},{78,8},{82,8},{83,8},{84,8},{85,8},{91,8},{92,8},{93,8},{94,8},{97,8},{98,8},{99,8},{100,8},{101,8},{104,8},{105,8},{106,8},{107,8},{108,8},{109,8},{113,8},{114,8},{115,8},{116,8},{120,8},{121,8},{122,8},{123,8},{3,9},{4,9},{5,9},{6,9},{23,9},{24,9},{25,9},{26,9},{32,9},{33,9},{34,9},{35,9},{38,9},{39,9},{40,9},{41,9},{42,9},{47,9},{48,9},{49,9},{50,9},{54,9},{55,9},{56,9},{57,9},{61,9},{62,9},{63,9},{64,9},{67,9},{68,9},{69,9},{70,9},{71,9},{76,9},{77,9},{78,9},{79,9},{82,9},{83,9},{84,9},{85,9},{91,9},{92,9},{93,9},{94,9},{97,9},{98,9},{99,9},{100,9},{106,9},{107,9},{108,9},{109,9},{113,9},{114,9},{115,9},{116,9},{120,9},{121,9},{122,9},{123,9},{3,10},{4,10},{5,10},{6,10},{23,10},{24,10},{25,10},{26,10},{32,10},{33,10},{34,10},{35,10},{38,10},{39,10},{40,10},{41,10},{47,10},{48,10},{49,10},{50,10},{54,10},{55,10},{56,10},{57,10},{61,10},{62,10},{63,10},{64,10},{67,10},{68,10},{69,10},{70,10},{76,10},{77,10},{78,10},{79,10},{82,10},{83,10},{84,10},{85,10},{91,10},{92,10},{93,10},{94,10},{106,10},{107,10},{108,10},{109,10},{113,10},{114,10},{115,10},{116,10},{120,10},{121,10},{122,10},{123,10},{3,11},{4,11},{5,11},{6,11},{22,11},{23,11},{24,11},{25,11},{33,11},{34,11},{35,11},{36,11},{38,11},{39,11},{40,11},{41,11},{47,11},{48,11},{49,11},{50,11},{54,11},{55,11},{56,11},{57,11},{61,11},{62,11},{63,11},{64,11},{67,11},{68,11},{69,11},{70,11},{76,11},{77,11},{78,11},{79,11},{82,11},{83,11},{84,11},{85,11},{91,11},{92,11},{93,11},{94,11},{102,11},{103,11},{104,11},{105,11},{106,11},{107,11},{108,11},{109,11},{113,11},{114,11},{115,11},{116,11},{120,11},{121,11},{122,11},{123,11},{3,12},{4,12},{5,12},{6,12},{22,12},{23,12},{24,12},{25,12},{33,12},{34,12},{35,12},{36,12},{38,12},{39,12},{40,12},{41,12},{47,12},{48,12},{49,12},{50,12},{54,12},{55,12},{56,12},{57,12},{61,12},{62,12},{63,12},{64,12},{67,12},{68,12},{69,12},{70,12},{76,12},{77,12},{78,12},{79,12},{82,12},{83,12},{84,12},{85,12},{91,12},{92,12},{93,12},{94,12},{99,12},{100,12},{101,12},{102,12},{103,12},{104,12},{105,12},{106,12},{107,12},{108,12},{109,12},{113,12},{114,12},{115,12},{116,12},{120,12},{121,12},{122,12},{123,12},{3,13},{4,13},{5,13},{6,13},{22,13},{23,13},{24,13},{25,13},{33,13},{34,13},{35,13},{36,13},{38,13},{39,13},{40,13},{41,13},{47,13},{48,13},{49,13},{50,13},{54,13},{55,13},{56,13},{57,13},{61,13},{62,13},{63,13},{64,13},{67,13},{68,13},{69,13},{70,13},{76,13},{77,13},{78,13},{79,13},{82,13},{83,13},{84,13},{85,13},{91,13},{92,13},{93,13},{94,13},{98,13},{99,13},{100,13},{101,13},{102,13},{103,13},{104,13},{105,13},{106,13},{107,13},{108,13},{109,13},{113,13},{114,13},{115,13},{116,13},{120,13},{121,13},{122,13},{123,13},{3,14},{4,14},{5,14},{6,14},{18,14},{19,14},{20,14},{21,14},{22,14},{23,14},{24,14},{25,14},{33,14},{34,14},{35,14},{36,14},{38,14},{39,14},{40,14},{41,14},{47,14},{48,14},{49,14},{50,14},{54,14},{55,14},{56,14},{57,14},{61,14},{62,14},{63,14},{64,14},{67,14},{68,14},{69,14},{70,14},{76,14},{77,14},{78,14},{79,14},{82,14},{83,14},{84,14},{85,14},{91,14},{92,14},{93,14},{94,14},{97,14},{98,14},{99,14},{100,14},{101,14},{106,14},{107,14},{108,14},{109,14},{113,14},{114,14},{115,14},{116,14},{120,14},{121,14},{122,14},{123,14},{4,15},{5,15},{6,15},{17,15},{18,15},{19,15},{20,15},{22,15},{23,15},{24,15},{25,15},{33,15},{34,15},{35,15},{36,15},{38,15},{39,15},{40,15},{41,15},{47,15},{48,15},{49,15},{50,15},{54,15},{55,15},{56,15},{57,15},{61,15},{62,15},{63,15},{64,15},{67,15},{68,15},{69,15},{70,15},{76,15},{77,15},{78,15},{79,15},{82,15},{83,15},{84,15},{85,15},{91,15},{92,15},{93,15},{94,15},{96,15},{97,15},{98,15},{99,15},{106,15},{107,15},{108,15},{109,15},{113,15},{114,15},{115,15},{116,15},{120,15},{121,15},{122,15},{123,15},{4,16},{5,16},{6,16},{7,16},{17,16},{18,16},{19,16},{20,16},{23,16},{24,16},{25,16},{26,16},{32,16},{33,16},{34,16},{35,16},{38,16},{39,16},{40,16},{41,16},{47,16},{48,16},{49,16},{50,16},{54,16},{55,16},{56,16},{57,16},{61,16},{62,16},{63,16},{64,16},{67,16},{68,16},{69,16},{70,16},{76,16},{77,16},{78,16},{79,16},{82,16},{83,16},{84,16},{85,16},{91,16},{92,16},{93,16},{94,16},{96,16},{97,16},{98,16},{99,16},{105,16},{106,16},{107,16},{108,16},{109,16},{113,16},{114,16},{115,16},{116,16},{120,16},{121,16},{122,16},{123,16},{5,17},{6,17},{7,17},{8,17},{16,17},{17,17},{18,17},{19,17},{23,17},{24,17},{25,17},{26,17},{32,17},{33,17},{34,17},{35,17},{38,17},{39,17},{40,17},{41,17},{47,17},{48,17},{49,17},{50,17},{54,17},{55,17},{56,17},{57,17},{61,17},{62,17},{63,17},{64,17},{67,17},{68,17},{69,17},{70,17},{76,17},{77,17},{78,17},{79,17},{82,17},{83,17},{84,17},{85,17},{90,17},{91,17},{92,17},{93,17},{94,17},{96,17},{97,17},{98,17},{99,17},{105,17},{106,17},{107,17},{108,17},{109,17},{113,17},{114,17},{115,17},{116,17},{120,17},{121,17},{122,17},{123,17},{5,18},{6,18},{7,18},{8,18},{9,18},{10,18},{14,18},{15,18},{16,18},{17,18},{18,18},{19,18},{24,18},{25,18},{26,18},{27,18},{28,18},{30,18},{31,18},{32,18},{33,18},{34,18},{38,18},{39,18},{40,18},{41,18},{47,18},{48,18},{49,18},{50,18},{54,18},{55,18},{56,18},{57,18},{61,18},{62,18},{63,18},{64,18},{67,18},{68,18},{69,18},{70,18},{76,18},{77,18},{78,18},{79,18},{83,18},{84,18},{85,18},{86,18},{87,18},{89,18},{90,18},{91,18},{92,18},{93,18},{94,18},{97,18},{98,18},{99,18},{100,18},{101,18},{103,18},{104,18},{105,18},{106,18},{107,18},{108,18},{109,18},{113,18},{114,18},{115,18},{116,18},{120,18},{121,18},{122,18},{123,18},{6,19},{7,19},{8,19},{9,19},{10,19},{11,19},{12,19},{13,19},{14,19},{15,19},{16,19},{17,19},{18,19},{24,19},{25,19},{26,19},{27,19},{28,19},{29,19},{30,19},{31,19},{32,19},{33,19},{34,19},{38,19},{39,19},{40,19},{41,19},{47,19},{48,19},{49,19},{50,19},{54,19},{55,19},{56,19},{57,19},{58,19},{59,19},{61,19},{62,19},{63,19},{64,19},{67,19},{68,19},{69,19},{70,19},{76,19},{77,19},{78,19},{79,19},{83,19},{84,19},{85,19},{86,19},{87,19},{88,19},{89,19},{90,19},{91,19},{92,19},{93,19},{94,19},{97,19},{98,19},{99,19},{100,19},{101,19},{102,19},{103,19},{104,19},{105,19},{106,19},{107,19},{108,19},{109,19},{113,19},{114,19},{115,19},{116,19},{117,19},{118,19},{120,19},{121,19},{122,19},{123,19},{8,20},{9,20},{10,20},{11,20},{12,20},{13,20},{14,20},{15,20},{16,20},{26,20},{27,20},{28,20},{29,20},{30,20},{31,20},{32,20},{38,20},{39,20},{40,20},{41,20},{47,20},{48,20},{49,20},{50,20},{55,20},{56,20},{57,20},{58,20},{59,20},{61,20},{62,20},{63,20},{64,20},{67,20},{68,20},{69,20},{70,20},{76,20},{77,20},{78,20},{79,20},{84,20},{85,20},{86,20},{87,20},{88,20},{89,20},{90,20},{91,20},{92,20},{93,20},{94,20},{98,20},{99,20},{100,20},{101,20},{102,20},{103,20},{104,20},{105,20},{106,20},{107,20},{108,20},{109,20},{114,20},{115,20},{116,20},{117,20},{118,20},{120,20},{121,20},{122,20},{123,20},

};

int marime[336][2]{

{30,0},{31,0},{32,0},{3,3},{4,3},{5,3},{6,3},{7,3},{8,3},{9,3},{10,3},{11,3},{12,3},{13,3},{14,3},{18,3},{19,3},{20,3},{21,3},{22,3},{25,3},{26,3},{27,3},{28,3},{29,3},{30,3},{31,3},{32,3},{34,3},{35,3},{36,3},{37,3},{38,3},{39,3},{40,3},{41,3},{42,3},{43,3},{44,3},{45,3},{49,3},{50,3},{51,3},{52,3},{53,3},{3,4},{4,4},{5,4},{6,4},{8,4},{9,4},{10,4},{11,4},{13,4},{14,4},{15,4},{17,4},{18,4},{21,4},{22,4},{23,4},{25,4},{26,4},{27,4},{28,4},{30,4},{31,4},{32,4},{34,4},{35,4},{36,4},{37,4},{39,4},{40,4},{41,4},{42,4},{44,4},{45,4},{46,4},{48,4},{49,4},{50,4},{52,4},{53,4},{54,4},{3,5},{4,5},{5,5},{8,5},{9,5},{10,5},{13,5},{14,5},{15,5},{16,5},{17,5},{18,5},{21,5},{22,5},{23,5},{25,5},{26,5},{27,5},{30,5},{31,5},{32,5},{34,5},{35,5},{36,5},{39,5},{40,5},{41,5},{44,5},{45,5},{46,5},{47,5},{48,5},{49,5},{53,5},{54,5},{55,5},{3,6},{4,6},{5,6},{8,6},{9,6},{10,6},{13,6},{14,6},{15,6},{20,6},{21,6},{22,6},{23,6},{25,6},{26,6},{27,6},{30,6},{31,6},{32,6},{34,6},{35,6},{36,6},{39,6},{40,6},{41,6},{44,6},{45,6},{46,6},{47,6},{48,6},{49,6},{53,6},{54,6},{55,6},{3,7},{4,7},{5,7},{8,7},{9,7},{10,7},{13,7},{14,7},{15,7},{17,7},{18,7},{19,7},{20,7},{21,7},{22,7},{23,7},{25,7},{26,7},{27,7},{30,7},{31,7},{32,7},{34,7},{35,7},{36,7},{39,7},{40,7},{41,7},{44,7},{45,7},{46,7},{47,7},{48,7},{49,7},{50,7},{51,7},{52,7},{53,7},{54,7},{55,7},{3,8},{4,8},{5,8},{8,8},{9,8},{10,8},{13,8},{14,8},{15,8},{16,8},{17,8},{18,8},{21,8},{22,8},{23,8},{25,8},{26,8},{27,8},{30,8},{31,8},{32,8},{34,8},{35,8},{36,8},{39,8},{40,8},{41,8},{44,8},{45,8},{46,8},{47,8},{48,8},{49,8},{3,9},{4,9},{5,9},{8,9},{9,9},{10,9},{13,9},{14,9},{15,9},{16,9},{17,9},{18,9},{21,9},{22,9},{23,9},{25,9},{26,9},{27,9},{30,9},{31,9},{32,9},{34,9},{35,9},{36,9},{39,9},{40,9},{41,9},{44,9},{45,9},{46,9},{47,9},{48,9},{49,9},{3,10},{4,10},{5,10},{8,10},{9,10},{10,10},{13,10},{14,10},{15,10},{16,10},{17,10},{18,10},{20,10},{21,10},{22,10},{23,10},{25,10},{26,10},{27,10},{30,10},{31,10},{32,10},{34,10},{35,10},{36,10},{39,10},{40,10},{41,10},{44,10},{45,10},{46,10},{48,10},{49,10},{50,10},{53,10},{54,10},{55,10},{3,11},{4,11},{5,11},{8,11},{9,11},{10,11},{13,11},{14,11},{15,11},{17,11},{18,11},{19,11},{20,11},{21,11},{22,11},{23,11},{24,11},{25,11},{26,11},{27,11},{30,11},{31,11},{32,11},{34,11},{35,11},{36,11},{39,11},{40,11},{41,11},{44,11},{45,11},{46,11},{49,11},{50,11},{51,11},{52,11},{53,11},{54,11},

};

int orientare[411][2]{

{6,3},{7,3},{8,3},{9,3},{10,3},{11,3},{20,3},{21,3},{22,3},{42,3},{43,3},{5,4},{6,4},{7,4},{10,4},{11,4},{12,4},{41,4},{42,4},{43,4},{4,5},{5,5},{6,5},{11,5},{12,5},{13,5},{41,5},{42,5},{43,5},{4,6},{5,6},{12,6},{13,6},{14,6},{15,6},{16,6},{17,6},{18,6},{19,6},{20,6},{21,6},{22,6},{26,6},{27,6},{28,6},{29,6},{30,6},{33,6},{34,6},{35,6},{36,6},{37,6},{38,6},{39,6},{40,6},{41,6},{42,6},{43,6},{44,6},{45,6},{47,6},{48,6},{49,6},{50,6},{51,6},{54,6},{55,6},{56,6},{57,6},{58,6},{59,6},{61,6},{62,6},{63,6},{64,6},{65,6},{3,7},{4,7},{5,7},{12,7},{13,7},{14,7},{15,7},{16,7},{17,7},{18,7},{20,7},{21,7},{22,7},{25,7},{26,7},{27,7},{29,7},{30,7},{31,7},{33,7},{34,7},{35,7},{36,7},{38,7},{39,7},{40,7},{41,7},{42,7},{43,7},{46,7},{47,7},{50,7},{51,7},{52,7},{54,7},{55,7},{56,7},{57,7},{60,7},{61,7},{62,7},{64,7},{65,7},{66,7},{3,8},{4,8},{5,8},{12,8},{13,8},{14,8},{15,8},{16,8},{17,8},{20,8},{21,8},{22,8},{24,8},{25,8},{26,8},{30,8},{31,8},{32,8},{33,8},{34,8},{35,8},{38,8},{39,8},{40,8},{41,8},{42,8},{43,8},{45,8},{46,8},{47,8},{50,8},{51,8},{52,8},{54,8},{55,8},{56,8},{59,8},{60,8},{61,8},{65,8},{66,8},{67,8},{3,9},{4,9},{5,9},{12,9},{13,9},{14,9},{15,9},{16,9},{17,9},{20,9},{21,9},{22,9},{24,9},{25,9},{26,9},{30,9},{31,9},{32,9},{33,9},{34,9},{35,9},{38,9},{39,9},{40,9},{41,9},{42,9},{43,9},{49,9},{50,9},{51,9},{52,9},{54,9},{55,9},{56,9},{59,9},{60,9},{61,9},{65,9},{66,9},{67,9},{3,10},{4,10},{5,10},{12,10},{13,10},{14,10},{15,10},{16,10},{17,10},{20,10},{21,10},{22,10},{24,10},{25,10},{26,10},{27,10},{28,10},{29,10},{30,10},{31,10},{32,10},{33,10},{34,10},{35,10},{38,10},{39,10},{40,10},{41,10},{42,10},{43,10},{46,10},{47,10},{48,10},{49,10},{50,10},{51,10},{52,10},{54,10},{55,10},{56,10},{59,10},{60,10},{61,10},{62,10},{63,10},{64,10},{65,10},{66,10},{67,10},{4,11},{5,11},{12,11},{13,11},{14,11},{15,11},{16,11},{17,11},{20,11},{21,11},{22,11},{24,11},{25,11},{26,11},{33,11},{34,11},{35,11},{38,11},{39,11},{40,11},{41,11},{42,11},{43,11},{45,11},{46,11},{47,11},{50,11},{51,11},{52,11},{54,11},{55,11},{56,11},{59,11},{60,11},{61,11},{4,12},{5,12},{6,12},{11,12},{12,12},{13,12},{15,12},{16,12},{17,12},{20,12},{21,12},{22,12},{24,12},{25,12},{26,12},{33,12},{34,12},{35,12},{38,12},{39,12},{40,12},{41,12},{42,12},{43,12},{45,12},{46,12},{47,12},{50,12},{51,12},{52,12},{54,12},{55,12},{56,12},{59,12},{60,12},{61,12},{5,13},{6,13},{7,13},{10,13},{11,13},{12,13},{15,13},{16,13},{17,13},{20,13},{21,13},{22,13},{25,13},{26,13},{27,13},{30,13},{31,13},{32,13},{33,13},{34,13},{35,13},{38,13},{39,13},{40,13},{41,13},{42,13},{43,13},{45,13},{46,13},{47,13},{49,13},{50,13},{51,13},{52,13},{54,13},{55,13},{56,13},{60,13},{61,13},{62,13},{65,13},{66,13},{67,13},{6,14},{7,14},{8,14},{9,14},{10,14},{11,14},{15,14},{16,14},{17,14},{20,14},{21,14},{22,14},{26,14},{27,14},{28,14},{29,14},{30,14},{31,14},{33,14},{34,14},{35,14},{38,14},{39,14},{40,14},{42,14},{43,14},{44,14},{45,14},{46,14},{47,14},{48,14},{49,14},{50,14},{51,14},{52,14},{53,14},{54,14},{55,14},{56,14},{61,14},{62,14},{63,14},{64,14},{65,14},{66,14},

};

bool limit(int x, int y) {

return (x >= 85 && x <= 1285 && y >= 75 && y <= 725) ? 1 : 0;

}

void \_brush(int x, int y, COLORREF color) {

for (int d = 0; d < 360; d++) {

for (int r = 0; r <= paint\_brush.size; r++) {

if (limit(x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180))))

SetPixel(consoleDC, x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180)), color);

}

}

}

void \_eraser(int x, int y) {

for (int d = 0; d < 360; d++) {

for (int r = 0; r <= eraser.size; r++) {

if (limit(x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180))))

SetPixel(consoleDC, x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180)), RGB(255, 255, 255));

}

}

}

int \_bucket(int f\_x, int f\_y, COLORREF start\_color, BOOL LIMIT) {

if (LIMIT) {

int bx[] = { 0,0,1,-1,1,-1,1,-1 };

int by[] = { 1,-1,0,0,-1,1,1,-1 };

int d[] = { 90,180,270,360 };

COLORREF color = GetPixel(consoleDC, f\_x, f\_y);

if (color == start\_color)

return 0;

vector<pair<int, int>> v;

v.push\_back(make\_pair(f\_x, f\_y));

SetPixel(consoleDC, f\_x, f\_y, start\_color);

int p = 0;

while (p < v.size()) {

for (int i = 0; i < 4; i++) {

if (GetPixel(consoleDC, v[p].first + bx[4 + i], v[p].second + by[4 + i]) == color && limit(v[p].first + bx[4 + i], v[p].second + by[4 + i]))

v.push\_back(make\_pair(v[p].first + bx[4 + i], v[p].second + by[4 + i])), SetPixel(consoleDC, v[p].first + bx[4 + i], v[p].second + by[4 + i], start\_color);

}

p++;

}

p = 0;

while (p < v.size()) {

for (int i = 0; i < 4; i++) {

if (GetPixel(consoleDC, v[p].first + bx[i], v[p].second + by[i]) == color)

SetPixel(consoleDC, v[p].first + bx[i], v[p].second + by[i], start\_color);

}

p++;

}

return 0;

}

else {

int bx[] = { 0,0,1,-1,1,-1,1,-1 };

int by[] = { 1,-1,0,0,-1,1,1,-1 };

int d[] = { 90,180,270,360 };

COLORREF color = GetPixel(consoleDC, f\_x, f\_y);

if (color == start\_color)

return 0;

vector<pair<int, int>> v;

v.push\_back(make\_pair(f\_x, f\_y));

SetPixel(consoleDC, f\_x, f\_y, start\_color);

int p = 0;

while (p < v.size()) {

for (int i = 0; i < 4; i++) {

if (GetPixel(consoleDC, v[p].first + bx[4 + i], v[p].second + by[4 + i]) == color)

v.push\_back(make\_pair(v[p].first + bx[4 + i], v[p].second + by[4 + i])), SetPixel(consoleDC, v[p].first + bx[4 + i], v[p].second + by[4 + i], start\_color);

}

p++;

}

p = 0;

while (p < v.size()) {

for (int i = 0; i < 4; i++) {

if (GetPixel(consoleDC, v[p].first + bx[i], v[p].second + by[i]) == color)

SetPixel(consoleDC, v[p].first + bx[i], v[p].second + by[i], start\_color);

}

p++;

}

return 0;

}

}

void \_pen(int x, int y, COLORREF color) {

for (int i = -pen.size; i <= pen.size; i++) {

if (limit(x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180))))

SetPixel(consoleDC, x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180)), color);

}

}

void \_line(double x1, double y1, double x2, double y2, COLORREF color) {

double d = sqrt((((x2 - x1) \* (x2 - x1)) + ((y2 - y1) \* (y2 - y1))));

double degree = atan2(y2 - y1, x2 - x1);

for (int i = 0; i <= d; i++) {

for (int j = -line.size; j <= line.size; j++) {

SetPixel(consoleDC, x1 + (i \* cos(degree)) + (j \* cos(degree + 90)), y1 + (i \* sin(degree)) + (j \* sin(degree + 90)), color);

}

}

}

double intensity = 1;

double cr, cg, cb;

//--FLOW--

/\*\_brush(x, y, RGB(intensity < 255 / 2 ? cr / (255 / 2) \* intensity : cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)),

intensity < 255 / 2 ? cg / (255 / 2) \* intensity : cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)),

intensity < 255 / 2 ? cb / (255 / 2) \* intensity : cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2)))), intensity++;

\_pen(x, y, RGB(cr + (r \* intensity), cg + (g \* intensity), cb + (b \* intensity))), intensity += 0.01;

if (intensity > 255)

intensity = 0;\*/

queue<pixel> q, r\_p, g\_p, b\_p, obscurity, ccp, b\_size, b\_orientation;

void draw\_cursor() {

int x = mouse\_pos.x, y = mouse\_pos.y;

if (brush\_t == "paint\_brush") {

int r = paint\_brush.size;

for (int d = 0; d < 360; d += 20) {

if (limit(x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180)))) {

q.push({ x + (r \* cos(d \* pi / 180)),y + (r \* sin(d \* pi / 180)),

GetPixel(consoleDC,x + (r \* cos(d \* pi / 180)),y + (r \* sin(d \* pi / 180))) });

SetPixel(consoleDC, mouse\_pos.x + (r \* cos(d \* pi / 180)), mouse\_pos.y + (r \* sin(d \* pi / 180)), RGB(125, 125, 125));

}

}

}

else if (brush\_t == "eraser") {

int r = eraser.size;

for (int d = 0; d < 360; d += 20) {

if (limit(x + (r \* cos(d \* pi / 180)), y + (r \* sin(d \* pi / 180)))) {

q.push({ x + (r \* cos(d \* pi / 180)),y + (r \* sin(d \* pi / 180)),

GetPixel(consoleDC,x + (r \* cos(d \* pi / 180)),y + (r \* sin(d \* pi / 180))) });

SetPixel(consoleDC, mouse\_pos.x + (r \* cos(d \* pi / 180)), mouse\_pos.y + (r \* sin(d \* pi / 180)), RGB(125, 125, 125));

}

}

}

else if (brush\_t == "pen") {

for (int i = -pen.size; i <= pen.size; i += 5) {

if (limit(x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180)))) {

q.push({ x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180)),

GetPixel(consoleDC,x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180))) });

SetPixel(consoleDC, x + (i \* cos(pen.orientation \* pi / 180)), y + (i \* sin(pen.orientation \* pi / 180)), RGB(125, 125, 125));

}

}

}

}

void clear\_pattern(queue<pixel>& q) {

while (!q.empty()) {

SetPixel(consoleDC, q.front().x, q.front().y, q.front().color);

q.pop();

}

}

void display\_cursor() {

clear\_pattern(q);

draw\_cursor();

}

void set\_indicator(queue<pixel>& q, double x, double y) {

for (int i = -1; i < 16; i++) {

q.push({ x,y + i,GetPixel(consoleDC,x,y + i) });

SetPixel(consoleDC, x, y + i, th == 0 ? RGB(0, 0, 0) : RGB(255, 255, 255));

}

}

void show\_pointer(queue<pixel>& q, double x, double y) {

for (int i = 0; i < 360; i += 20) {

q.push({ x + (5 \* cos(i \* pi / 180)),y + (5 \* sin(i \* pi / 180)),GetPixel(consoleDC,x + (5 \* cos(i \* pi / 180)),y + (5 \* sin(i \* pi / 180))) });

SetPixel(consoleDC, x + (5 \* cos(i \* pi / 180)), y + (5 \* sin(i \* pi / 180)), RGB(255, 255, 255));

}

}

void color\_indicator(COLORREF color) {

for (int d = 0; d < 360; d++) {

for (int r = 0; r <= 11; r++) {

SetPixel(consoleDC, 1450 + (r \* cos(d \* pi / 180)), 156 + (r \* sin(d \* pi / 180)), color);

}

}

}

void brush\_type(int x, int y) {

if (brush\_t == "paint\_brush") {

\_brush(x, y, RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (brush\_t == "bucket")

\_bucket(x, y, RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value), true);

else if (brush\_t == "pen")

\_pen(x, y, RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

else if (brush\_t == "line") {

double x2, y2;

this\_thread::sleep\_for(chrono::milliseconds(100));

while (true) {

GetCursorPos(&mouse\_pos);

ScreenToClient(consoleWindow, &mouse\_pos);

if ((GetAsyncKeyState((unsigned short)VK\_RBUTTON) & 0x8000) && limit(mouse\_pos.x, mouse\_pos.y)) {

x2 = mouse\_pos.x;

y2 = mouse\_pos.y;

break;

}

}

\_line(x, y, x2, y2, RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

this\_thread::sleep\_for(chrono::milliseconds(300));

}

else if (brush\_t == "eraser")

\_eraser(x, y);

else if (brush\_t == "picker") {

COLORREF local\_color = GetPixel(consoleDC, x, y);

r = GetRValue(local\_color);

g = GetGValue(local\_color);

b = GetBValue(local\_color);

clear\_pattern(r\_p);

set\_indicator(r\_p, 1305 + (r / 1.7), 209);

clear\_pattern(g\_p);

set\_indicator(g\_p, 1305 + (g / 1.7), 239);

clear\_pattern(b\_p);

set\_indicator(b\_p, 1305 + (b / 1.7), 269);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

}

void display\_color\_palette(int pos\_x, int pos\_y) {

int r = 0;

for (double intensity = 255; intensity > 133; intensity -= 1.5, r += 1) {

float degree = 0;

cr = 255, cg = 0, cb = 0;

for (degree; degree < 60; degree += 0.5, cg += 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

for (degree; degree < 120; degree += 0.5, cr -= 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

for (degree; degree < 180; degree += 0.5, cb += 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

for (degree; degree < 240; degree += 0.5, cg -= 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

for (degree; degree < 300; degree += 0.5, cr += 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

for (degree; degree < 360; degree += 0.5, cb -= 2.125) {

SetPixel(consoleDC, pos\_x + (r \* cos(degree \* pi / 180)), pos\_y + (r \* sin(degree \* pi / 180)), RGB(cr + (255 - cr) / (255 / 2) \* (intensity - (255 / 2)), cg + (255 - cg) / (255 / 2) \* (intensity - (255 / 2)), cb + (255 - cb) / (255 / 2) \* (intensity - (255 / 2))));

}

}

}

void draw(int x, int y) {

brush\_type(x, y);

};

void ColorPaletteUI(COLORREF body, COLORREF bezels) {

int origin\_x = 1375, origin\_y = 85;

/// COLOR RGB(0, 0, 13)

for (int i = 0; i < 835; i++) {

for (int x = origin\_x - 87; x < 1500; x++) {

SetPixel(consoleDC, x, origin\_y - 108 + i, body);

}

SetPixel(consoleDC, origin\_x - 88, origin\_y - 108 + i, bezels);

SetPixel(consoleDC, origin\_x - 89, origin\_y - 108 + i, bezels);

}

display\_color\_palette(origin\_x + 5, origin\_y);

for (int i = 0; i < 2; i++)

for (float d = 0; d < 360; d += 0.5) {

SetPixel(consoleDC, origin\_x + 5 + ((81 - i) \* cos(d \* pi / 180)), origin\_y + ((81 - i) \* sin(d \* pi / 180)), bezels);

}

for (double d = 0; d < 360; d += 0.5) {

SetPixel(consoleDC, origin\_x + 75 + (12 \* cos(d \* pi / 180)), origin\_y - 4 + 75 + (12 \* sin(d \* pi / 180)), bezels);

}

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

// Red

for (int lvl = 0; lvl < 15; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, origin\_x - 70 + i, origin\_y + 125 + lvl, RGB(1.7 \* i, 0, 0));

}

}

// Green

for (int lvl = 0; lvl < 15; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, origin\_x - 70 + i, origin\_y + 155 + lvl, RGB(0, 1.7 \* i, 0));

}

}

// Blue

for (int lvl = 0; lvl < 15; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, origin\_x - 70 + i, origin\_y + 185 + lvl, RGB(0, 0, 1.7 \* i));

}

}

//Shade

for (int lvl = 0; lvl < 15; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, origin\_x - 70 + 150 - i, origin\_y + 215 + lvl, RGB(1.7 \* i, 1.7 \* i, 1.7 \* i));

}

}

}

void tool\_bar(COLORREF body, COLORREF bezels) {

for (int i = 0; i < 835; i++) {

for (int x = 0; x < 35; x++) {

SetPixel(consoleDC, x, i, body);

}

SetPixel(consoleDC, 35, i, bezels);

SetPixel(consoleDC, 36, i, bezels);

}

}

COLORREF color[5][2] = {

{ RGB(99,98,90),RGB(68,0,204)},

{ RGB(0,0,0),RGB(0,0,0)},

{ RGB(140,145,127),RGB(34,8,127)},

{ RGB(200,200,200),RGB(25,25,25)},

{ RGB(73, 136, 245),RGB(112, 54, 199)},

};

void top\_bar(COLORREF body, COLORREF bezels) {

for (int y = 0; y < 29; y++) {

for (int x = 0; x < 1286; x++) {

SetPixel(consoleDC, x, y, body);

}

}

for (int x = 0; x < 1288; x++) {

SetPixel(consoleDC, x, 29, bezels);

}

for (int x = 0; x < 1288; x++) {

SetPixel(consoleDC, x, 30, bezels);

}

//size

for (int lvl = 0; lvl < 10; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, 600 - 70 + i, 10 + lvl, color[4][th]);

}

}

for (int i = 0; i <= 336; i++) {

SetPixel(consoleDC, 600 - 140 + marime[i][0], 8 + marime[i][1], th == 0 ? RGB(0, 0, 0) : RGB(255, 255, 255));

}

//orientation

for (int lvl = 0; lvl < 10; lvl++) {

for (int i = 150; i >= 0; i--) {

SetPixel(consoleDC, 600 + 270 - 70 + i, 10 + lvl, color[4][th]);

}

}

for (int i = 0; i <= 410; i++) {

SetPixel(consoleDC, 600 + 270 - 140 + orientare[i][0], 4 + orientare[i][1], th == 0 ? RGB(0, 0, 0) : RGB(255, 255, 255));

}

}

void initialize\_canvas() {

int origin\_x = 85, origin\_y = 75;

for (int y = 0; y < 650; y++) {

for (int x = 0; x < 1200; x++) {

SetPixel(consoleDC, origin\_x + x, origin\_y + y, RGB(255, 255, 255));

}

}

}

void Theme() {

for (int y = 0; y < 500; y++) {

for (int x = 0; x < 800; x++) {

SetPixel(consoleDC, x, y, RGB(255, 255, 255));

}

}

for (int i = 0; i <= 5556; i++) {

SetPixel(consoleDC, 20 + intro[i][0], 20 + intro[i][1], RGB(0, 0, 0));

}

//

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x, 99 + 50, RGB(50, 50, 50));//

for (int y = 100 + 50; y < 300 + 50; y++) {//

SetPixel(consoleDC, 49, y, RGB(50, 50, 50));//

for (int x = 50; x < 350; x++) {//

SetPixel(consoleDC, x, y, RGB(0, 0, 0));//

}

SetPixel(consoleDC, 350, y, RGB(50, 50, 50));//

}

for (int x = 50; x < 350; x++)//

SetPixel(consoleDC, x, 300 + 50, RGB(50, 50, 50));//

//

for (int y = 120 + 50; y < 300 + 50; y++) {//

for (int x = 0; x < 20; x++)//

SetPixel(consoleDC, 50 + x, y, RGB(242, 242, 242));//

SetPixel(consoleDC, 50 + 20, y, RGB(200, 200, 200));//

SetPixel(consoleDC, 50 + 20 + 1, y, RGB(200, 200, 200));//

}

for (int x = 50; x < 275; x++) {//

for (int y = 0 + 50; y < 20 + 50; y++)//

SetPixel(consoleDC, x, 100 + y, RGB(242, 242, 242));

SetPixel(consoleDC, x, 120 + 50, RGB(200, 200, 200));//

SetPixel(consoleDC, x, 120 + 1 + 50, RGB(200, 200, 200));//

}

for (int y = 100 + 50; y < 300 + 50; y++) {//

for (int x = 0; x < 75; x++) {//

SetPixel(consoleDC, 275 + x, y, RGB(242, 242, 242));//

}//

SetPixel(consoleDC, 275, y, RGB(200, 200, 200));//

SetPixel(consoleDC, 275 + 1, y, RGB(200, 200, 200));//

}

//

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 99 + 50, RGB(50, 50, 50));

for (int y = 100 + 50; y < 300 + 50; y++) {

SetPixel(consoleDC, 49 + 375, y, RGB(50, 50, 50));

for (int x = 50; x < 350; x++) {

SetPixel(consoleDC, x + 375, y, RGB(0, 0, 0));

}

SetPixel(consoleDC, 350 + 375, y, RGB(50, 50, 50));

}

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 300 + 50, RGB(50, 50, 50));

//

for (int y = 120 + 50; y < 300 + 50; y++) {

for (int x = 0; x < 20; x++)

SetPixel(consoleDC, 375 + 50 + x, y, RGB(7, 9, 15));

SetPixel(consoleDC, 375 + 50 + 20, y, RGB(34, 45, 64));

SetPixel(consoleDC, 375 + 50 + 20 + 1, y, RGB(34, 45, 64));

}

for (int x = 50; x < 275; x++) {

for (int y = 0 + 50; y < 20 + 50; y++)

SetPixel(consoleDC, 375 + x, 100 + y, RGB(7, 9, 15));

SetPixel(consoleDC, 375 + x, 120 + 50, RGB(34, 45, 64));

SetPixel(consoleDC, 375 + x, 120 + 1 + 50, RGB(34, 45, 64));

}

for (int y = 100 + 50; y < 300 + 50; y++) {

for (int x = 0; x < 75; x++) {

SetPixel(consoleDC, 375 + 275 + x, y, RGB(7, 9, 15));

}

SetPixel(consoleDC, 375 + 275, y, RGB(34, 45, 64));

SetPixel(consoleDC, 375 + 275 + 1, y, RGB(34, 45, 64));

}

//

for (int r = 0; r < 20; r++) {

for (double d = 0; d < 360; d += 0.1) {

SetPixel(consoleDC, 725 + (r \* cos(d \* pi / 180)), 425 + (r \* sin(d \* pi / 180)), RGB(42, 116, 235));

}

}

for (int r = -13; r < 16; r++) {

SetPixel(consoleDC, 725 + r, 425, RGB(255, 255, 255));

}

for (double r = 0; r < 10; r += 0.5) {

SetPixel(consoleDC, 725 + 15 + (r \* cos(130 \* pi / 180)), 425 + (r \* sin(130 \* pi / 180)), RGB(255, 255, 255));

}

for (double r = 0; r < 10; r += 0.5) {

SetPixel(consoleDC, 725 + 15 + (r \* cos(-130 \* pi / 180)), 425 + (r \* sin(-130 \* pi / 180)), RGB(255, 255, 255));

}

//

for (int i = 0; i <= 1134; i++) {

SetPixel(consoleDC, 575 + continuati[i][0], 415 + continuati[i][1], RGB(0, 0, 0));

}

}

int pb[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,2,2,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,1,1,3,3,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,1,1,1,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,1,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int bk[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,2,2,1,1,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,1,2,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,1,2,2,2,2,2,2,2,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,1,1,2,2,2,2,2,2,2,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,1,1,1,3,3,3,3,3,3,1,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,1,1,3,3,3,3,3,3,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,1,1,1,3,3,3,3,1,0,0,0,0,

0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,1,3,3,3,3,0,0,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,3,3,3,0,0,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,1,3,3,3,0,0,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,3,3,0,0,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,0,0,0,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,0,0,0,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,0,0,0,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,3,0,0,0,0,0,

0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,3,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,3,3,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,1,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,3,3,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,1,0,0,0,0,3,3,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,3,3,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int pn[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,3,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,3,3,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,3,3,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,3,3,3,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,3,3,3,3,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,1,1,1,1,1,1,1,3,3,3,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,1,1,1,1,3,3,3,3,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,1,1,1,3,3,3,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,2,2,2,2,3,3,3,3,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,2,2,2,2,0,2,2,2,3,3,3,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,2,2,2,0,0,2,2,2,2,2,3,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,2,0,0,2,2,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,2,0,2,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,2,2,2,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int fg[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,1,1,0,0,1,0,0,1,1,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int ln[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,1,0,0,2,2,2,2,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,1,0,2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,1,0,2,0,0,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,1,0,2,0,0,1,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,1,1,1,1,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,0,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,2,0,0,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,0,0,0,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,0,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int er[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,2,1,1,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,2,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,3,1,2,2,2,2,2,2,2,1,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,1,1,2,2,2,2,2,2,2,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,1,1,2,2,2,2,2,2,2,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,1,1,2,2,2,2,2,2,2,1,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,1,2,2,2,2,2,2,2,1,0,0,0,0,

0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,1,2,2,2,2,2,2,2,1,0,0,0,

0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,1,2,2,2,2,2,2,2,1,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,2,2,2,2,2,2,1,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,1,2,2,2,2,1,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,2,2,2,1,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,2,1,0,0,0,

0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,

0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,0,

0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,1,3,3,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,3,3,3,3,3,3,3,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

int pk[] = {

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,2,2,2,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,0,2,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,2,0,1,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,1,0,2,1,1,2,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,2,0,0,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,1,0,0,0,0,0,2,0,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,3,0,0,0,0,1,0,2,2,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,3,3,0,0,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,3,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,1,3,3,3,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,1,1,1,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

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,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,

};

void display\_icon(int v[32 \* 32], double x, double y) {

for (double d = 0; d < 360; d += 0.5) {

for (int r = 1; r <= 14; r++) {

SetPixel(consoleDC, x + 16 + (r \* cos(d \* pi / 180)), y + 16 + (r \* sin(d \* pi / 180)), color[3][th]);

}

}

for (int i = 0; i < 32; i++) {

for (int j = 0; j < 32; j++) {

switch (v[i \* 32 + j]) {

case 1:

SetPixel(consoleDC, x + j, y + i, color[0][th]);

break;

case 2:

SetPixel(consoleDC, x + j, y + i, color[1][th]);

break;

case 3:

SetPixel(consoleDC, x + j, y + i, color[2][th]);

break;

}

}

}

}

int main() {

COLORREF body = NULL, bezels = NULL;

RECT ConsoleRect;

GetWindowRect(consoleWindow, &ConsoleRect);

MoveWindow(consoleWindow, ConsoleRect.left, ConsoleRect.top, 800, 500, TRUE);

Theme();

while (true) {

GetCursorPos(&mouse\_pos);

ScreenToClient(consoleWindow, &mouse\_pos);

if (GetAsyncKeyState((unsigned short)VK\_RBUTTON) & 0x8000) {

if (sqrt((mouse\_pos.x - 725) \* (mouse\_pos.x - 725)) + ((mouse\_pos.y - 425) \* (mouse\_pos.y - 425)) <= 19 && (bezels != NULL && body != NULL)) {

break;

}

if (mouse\_pos.x > 49 && mouse\_pos.x < 350 && mouse\_pos.y>49 && mouse\_pos.y < 300) {

body = RGB(242, 242, 242), bezels = RGB(200, 200, 200), th = 0;

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x, 99 + 50, RGB(0, 153, 255));

for (int y = 100 + 50; y < 300 + 50; y++) {

SetPixel(consoleDC, 49, y, RGB(0, 153, 255));

SetPixel(consoleDC, 350, y, RGB(0, 153, 255));

}

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x, 300 + 50, RGB(0, 153, 255));

//

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 99 + 50, RGB(50, 50, 50));

for (int y = 100 + 50; y < 300 + 50; y++) {

SetPixel(consoleDC, 49 + 375, y, RGB(50, 50, 50));

SetPixel(consoleDC, 350 + 375, y, RGB(50, 50, 50));

}

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 300 + 50, RGB(50, 50, 50));

}

if (mouse\_pos.x > 424 && mouse\_pos.x < 725 && mouse\_pos.y>99 && mouse\_pos.y < 300) {

body = RGB(7, 9, 15), bezels = RGB(34, 45, 64), th = 1;

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 99 + 50, RGB(0, 153, 255));

for (int y = 100 + 50; y < 300 + 50; y++) {

SetPixel(consoleDC, 49 + 375, y, RGB(0, 153, 255));

SetPixel(consoleDC, 350 + 375, y, RGB(0, 153, 255));

}

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x + 375, 300 + 50, RGB(0, 153, 255));

//

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x, 99 + 50, RGB(50, 50, 50));

for (int y = 100 + 50; y < 300 + 50; y++) {

SetPixel(consoleDC, 49, y, RGB(50, 50, 50));

SetPixel(consoleDC, 350, y, RGB(50, 50, 50));

}

for (int x = 50; x < 350; x++)

SetPixel(consoleDC, x, 300 + 50, RGB(50, 50, 50));

}

}

}

MoveWindow(consoleWindow, ConsoleRect.left, ConsoleRect.top, 1500, 850, TRUE);

this\_thread::sleep\_for(chrono::seconds(1));

initialize\_canvas();

ColorPaletteUI(body, bezels);

tool\_bar(body, bezels);

top\_bar(body, bezels);

GetCursorPos(&mouse\_pos);

ScreenToClient(consoleWindow, &mouse\_pos);

set\_indicator(r\_p, 1305, 209);

set\_indicator(g\_p, 1305, 239);

set\_indicator(b\_p, 1305, 269);

set\_indicator(obscurity, 1305, 299);

set\_indicator(b\_size, 530, 8);

set\_indicator(b\_orientation, 800, 8);

display\_icon(pb, 3, 50);

\_bucket(11, 57, color[4][th], false);

display\_icon(bk, 3, 125);

display\_icon(pn, 3, 200);

display\_icon(ln, 3, 275);

display\_icon(fg, 3, 350);

display\_icon(er, 3, 425);

for (double d = 0; d < 360; d += 0.5) {

for (int r = 1; r <= 14; r++) {

SetPixel(consoleDC, 1290 + 16 + (r \* cos(d \* pi / 180)), 140 + 16 + (r \* sin(d \* pi / 180)), color[3][th]);

}

}

for (int i = 0; i < 32; i++) {

for (int j = 0; j < 32; j++) {

switch (pk[i \* 32 + j]) {

case 1:

SetPixel(consoleDC, 1290 + j, 140 + i, color[0][th]);

break;

case 2:

SetPixel(consoleDC, 1290 + j, 140 + i, color[1][th]);

break;

case 3:

SetPixel(consoleDC, 1290 + j, 140 + i, color[2][th]);

break;

}

}

}

int last = 0;

int x = mouse\_pos.x, y = mouse\_pos.y;

while (true) {

GetCursorPos(&mouse\_pos);

ScreenToClient(consoleWindow, &mouse\_pos);

if (GetAsyncKeyState((unsigned short)VK\_RBUTTON) & 0x8000) {

int pos\_x = mouse\_pos.x, pos\_y = mouse\_pos.y;

clear\_pattern(q);

if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (50 + 16), 2)) <= 16 && last != 0) {

brush\_t = "paint\_brush";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530 + (paint\_brush.size \* 3), 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(11, 57, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 0;

}

else if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (125 + 16), 2)) <= 16 && last != 1) {

brush\_t = "bucket";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530, 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(11, 134, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 1;

}

else if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (200 + 16), 2)) <= 16 && last != 2) {

brush\_t = "pen";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530 + (pen.size \* 3), 8);

set\_indicator(b\_orientation, 800 + (pen.orientation \* 0.416), 8);

\_bucket(11, 208, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 2;

}

else if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (275 + 16), 2)) <= 16 && last != 3) {

brush\_t = "line";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530 + (line.size \* 3), 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(17, 280, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 3;

}

else if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (350 + 16), 2)) <= 16 && last != 4) {

brush\_t = "figures";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530, 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(17, 353, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 4;

}

else if (sqrt(pow(mouse\_pos.x - (3 + 16), 2) + pow(mouse\_pos.y - (425 + 16), 2)) <= 16 && last != 4) {

brush\_t = "eraser";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530 + (eraser.size \* 3), 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(17, 428, color[4][th], false);

last == 6 ? \_bucket(1290 + 16, 140 + 25, color[3][th], false) : \_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 5;

}

//1290 + 16 140 + 16

else if (sqrt(pow(mouse\_pos.x - (1290 + 16), 2) + pow(mouse\_pos.y - (140 + 16), 2)) <= 14 && last != 6) {

brush\_t = "picker";

clear\_pattern(b\_size);

clear\_pattern(b\_orientation);

set\_indicator(b\_size, 530, 8);

set\_indicator(b\_orientation, 800, 8);

\_bucket(1290 + 16, 140 + 25, color[4][th], false);

\_bucket(15, 57 + (75 \* last), color[3][th], false);

last = 6;

}

if (limit(pos\_x, pos\_y))

draw(pos\_x, pos\_y);

else if (sqrt(((1380 - pos\_x) \* (1380 - pos\_x)) + ((85 - pos\_y) \* (85 - pos\_y))) <= 80) {

clear\_pattern(ccp);

COLORREF local\_color = GetPixel(consoleDC, pos\_x, pos\_y);

r = GetRValue(local\_color), g = GetGValue(local\_color), b = GetBValue(local\_color);

show\_pointer(ccp, pos\_x, pos\_y);

clear\_pattern(r\_p);

set\_indicator(r\_p, 1305 + (r / 1.7), 209);

clear\_pattern(g\_p);

set\_indicator(g\_p, 1305 + (g / 1.7), 239);

clear\_pattern(b\_p);

set\_indicator(b\_p, 1305 + (b / 1.7), 269);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (pos\_x >= 1305 && pos\_y >= 209 && pos\_x <= 1455 && pos\_y <= 224) {

//COLORREF local\_color = GetPixel(consoleDC, pos\_x, pos\_y);

//local\_color != RGB(255, 255, 255) ? r = GetRValue(local\_color):0;

r = (pos\_x - 1305) \* 1.7;

clear\_pattern(r\_p);

set\_indicator(r\_p, pos\_x, 209);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (pos\_x >= 1305 && pos\_y >= 239 && pos\_x <= 1455 && pos\_y <= 254) {

g = (pos\_x - 1305) \* 1.7;

clear\_pattern(g\_p);

set\_indicator(g\_p, pos\_x, 239);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (pos\_x >= 1305 && pos\_y >= 269 && pos\_x <= 1455 && pos\_y <= 284) {

b = (pos\_x - 1305) \* 1.7;

clear\_pattern(b\_p);

set\_indicator(b\_p, pos\_x, 269);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (pos\_x >= 1305 && pos\_y >= 299 && pos\_x <= 1455 && pos\_y <= 314) {

shade\_value = 100 - (0.66 \* (pos\_x - 1305));

clear\_pattern(obscurity);

set\_indicator(obscurity, pos\_x, 299);

color\_indicator(RGB(r / 100 \* shade\_value, g / 100 \* shade\_value, b / 100 \* shade\_value));

}

else if (pos\_x >= 530 && pos\_y >= 12 && pos\_x <= 680 && pos\_y <= 22) {

if (brush\_t == "paint\_brush")

paint\_brush.size = (pos\_x - 530) \* 0.33;

else if (brush\_t == "pen")

pen.size = (pos\_x - 530) \* 0.33;

else if (brush\_t == "line")

line.size = (pos\_x - 530) \* 0.33;

else if (brush\_t == "eraser")

eraser.size = (pos\_x - 530) \* 0.33;

clear\_pattern(b\_size);

set\_indicator(b\_size, pos\_x, 8);

}

else if (pos\_x >= 800 && pos\_y >= 12 && pos\_x <= 951 && pos\_y <= 22) {

if (brush\_t == "pen")

pen.orientation = (pos\_x - 800) \* 2.4;

clear\_pattern(b\_orientation);

set\_indicator(b\_orientation, pos\_x, 8);

}

}

else if ((x != mouse\_pos.x || y != mouse\_pos.y) && limit(mouse\_pos.x, mouse\_pos.y)) {

x = mouse\_pos.x, y = mouse\_pos.y;

display\_cursor();

}

}

ReleaseDC(consoleWindow, consoleDC);

cin.ignore();

}

**Bibliografie:**

**-Referințe cod:**

https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-getpixel

https://docs.microsoft.com/en-us/windows/win32/api/wingdi/nf-wingdi-setpixel

https://docs.microsoft.com/en-us/windows/win32/inputdev/virtual-key-codes

https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-getcursorpos

https://docs.microsoft.com/en-us/windows/win32/api/winuser/nf-winuser-screentoclient

https://stackoverflow.com/questions/12378642/c-pixels-in-console-window

Codul a fost realizat de **Boroica Marius** iar câteva părți grafice au fost făcute de **Ganea Alexandru**.