//#include<iostream>

//using namespace std;

// //programmer ram

//char\* mystrstr(char\* str1, const char\* str2) {

// int l1 = strlen(str1);

// int l2 = strlen(str2);

// for (size\_t i = 0; i < l1; i++)

// {

// bool hasExited = false;

// if (str1[i] == str2[0]) {

// int k2 = i;

// for (size\_t k = 0; k < l2; k++,k2++)

// {

// if (str2[k] != str1[k2]) {

// hasExited = true;

// break;

// }

// }

// if (hasExited) {

// i = k2;

// }

// else {

// return &str1[i];

// }

// }

// }

// return nullptr;

//}

//void main() {

//

// /\* char\* mystrstr (const char \*str1, const char \*str2); - funksiya str1

// sətrinin içində str2 sətrini axtarır. Tapdığı halda ilk

// hərfə pointer qaytarır, tapmazsa 0.\*/

// char\* text1 = new char[] {"Programmer"};

// char\* text2 = new char[] {"gra"};

// cout<<mystrstr(text1, text2)<<endl;

//

//}

// 8. char\* mynumtostr (int number); - funksiya rəqəmi

// sətirə çevirir və sətirə pointer qaytarır.

//void printArray(char\* arr, const int\* size)

//{

// cout << "\n My Char Array : ";

// for (int x = 0; x < \*size; x++)

// {

// cout << \*(arr + x) << ", ";

// }

// cout << "\b\b.";

//}

//void reverseArray(char\* arr, const int\* size)

//{

// for (int x = 0; x < \*size / 2; x++)

// {

// char temp = arr[x];

// arr[x] = arr[\*size - x - 1];

// arr[\*size - x - 1] = temp;

// }

//}

//int intlen(int number)

//{

// int counter = 0;

// while (number > 0)

// {

// number /= 10;

// counter++;

// }

// return counter;

//}

//char\* mynumtostr(int number)

//{

// int size = intlen(number);

// char\* arr = new char[size] {};

// int index = 0;

// while (number > 0)

// {

// int o = number % 10; //1+48=(char)49

// arr[index] = (char)(o + 48);

// ++index;

// number /= 10;

// }

// arr[size] = '\0';

// reverseArray(arr, &size);

// return arr;

//}

//

//

//

//void main() {

//

//

// cout << mynumtostr(11234) << endl;

//}

#include<iostream>

#include<conio.h>

#include<Windows.h>

using namespace std;

int X = 0;

int Y = 0;

const int s = 12;

int game[s][s] = {};

int live = 3;

bool hasStoped = false;

int score = 0;

void SetFruits();

void ShowGame() {

for (size\_t i = 0; i < s; i++)

{

for (size\_t k = 0; k < s; k++)

{

if (i == 0 || i == s - 1) {

cout << "- ";

}

else if (k == 0) {

cout << "| ";

}

else if (k == s - 1) {

cout << "|";

}

else if (game[i][k] == 5) {

HANDLE consolehwnd = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_RED);

cout << (char)209 << " ";

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

}

else if (game[i][k] == 2) {

HANDLE consolehwnd = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_GREEN);

cout << 'O' << " ";

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

}

else if (game[i][k] == 1) {

HANDLE consolehwnd = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_BLUE);

cout << (char)2 << " ";

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

}

else if (game[i][k] == 9) {

HANDLE consolehwnd = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

cout << (char)234 << " ";

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

}

else {

cout << " ";

}

}

cout << endl;

}

cout << "SCORE : " << score << " ";

for (size\_t i = 0; i < live; i++)

{

HANDLE consolehwnd = GetStdHandle(STD\_OUTPUT\_HANDLE);

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_RED);

cout << (char)3 << " ";

SetConsoleTextAttribute(consolehwnd, FOREGROUND\_INTENSITY);

}

}

void ResetMushrooms() {

for (size\_t i = 0; i < s-1; i++)

{

for (size\_t k = 0; k < s-1; k++)

{

if (game[i][k] == 9) {

game[i][k] = 0;

}

}

}

}

void SetWalls() {

game[3][3] = 5;

game[4][3] = 5;

game[5][3] = 5;

game[6][3] = 5;

game[6][4] = 5;

game[6][5] = 5;

game[6][2] = 5;

game[6][3] = 5;

game[6][4] = 5;

game[6][5] = 5;

game[6][6] = 5;

game[4][6] = 5;

game[5][6] = 5;

game[6][6] = 5;

}

bool isEmpty(int x, int y) {

if (x == 0 || y == 0 || x == s - 1 || y == s - 1) {

live--;

cout << "\a";

if (live == 0) {

hasStoped = true;

system("color 4");

}

return false;

}

if (game[y][x] == 5) {

return false;

}

else if (game[y][x] == 9) {

score -= 10;

}

else if (game[y][x] == 2) {

score += 10;

if (score == 50) {

ResetMushrooms();

}

if (score == 20) {

SetWalls();

}

SetFruits();

}

return true;

}

void StartLoop() {

enum Directions { UP = 72, DOWN = 80, RIGHT = 77, LEFT = 75 };

while (true)

{

system("cls");

ShowGame();

if (hasStoped) {

return;

}

char symbol = \_getch();

if (symbol == -32) {

symbol = \_getch();

int no = symbol;

if (no == UP) {

if (!isEmpty(X, Y - 1)) {

continue;

}

game[Y][X] = 0;

Y--;

}

else if (no == DOWN) {

if (!isEmpty(X, Y + 1)) {

continue;

}

game[Y][X] = 0;

Y++;

}

else if (no == LEFT) {

if (!isEmpty(X - 1, Y)) {

continue;

}

game[Y][X] = 0;

X--;

}

else if (no == RIGHT) {

if (!isEmpty(X + 1, Y)) {

continue;

}

game[Y][X] = 0;

X++;

}

game[Y][X] = 1;

}

}

}

void SetMushroom();

void SetFruits() {

while (true)

{

int rX = 1 + rand() % (9);

int rY = 1 + rand() % (9);

if (game[rY][rX] ==0) {

game[rY][rX] = 2;

break;

}

}

SetMushroom();

SetMushroom();

}

void SetMushroom() {

while (true)

{

int rX = 1 + rand() % (9);

int rY = 1 + rand() % (9);

if (game[rY][rX] == 0) {

game[rY][rX] = 9;

break;

}

}

}

void main() {

srand(time(0));

X = 1;

Y = 1;

game[Y][X] = 1;

SetFruits();

ShowGame();

StartLoop();

}