Задача 100

Подготовил Марченко С.И. КВБО-01-21

**Программа «Жизнь», (описывающая жизнь микроорганизмов)**

**Введение:**

Жили-были микробы. Жили они долго и счастливо, но вот одна беда: после жизни всегда приходит смерть. А микробы были разные: маленькие и не очень, старички и малютки – и у каждого из них был свой жизненный уровень. Так, например, у только что родившегося микробчика он был равен 1, а по мере его взросления жизненный уровень тоже рос (от 1 до 12). Когда же микроб достигал последней ступени (т.е. 12), он, увы, погибал (теперь его жизненный уровень равен 0). Если же у микроба уровень был равен 0, то он рождался заново и проходил опять жизнь бодрым шагом от 1 до 12. Всего же микробов в мире, как вы знаете, очень много, а жизнь их интересна. Поэтому появилась идея написать программу, описывающую их жизнь. Эта программа должна сообщать нам количество микробов в каждом поколении и «нарисовать» их на поле (экране).

**Программа должна работать следующим образом:**

1. Сначала создаётся файл work.dat (жилище микробов) и файл work.out (описывающий текущее поколение микробов, их развитие). Файл work.dat состоит из различных символов, среди которых «обитают» микробы.

2. Необходимо создать массивы «настоящее» и «будущее». В массиве «настоящее» записывается текущее поколение микробов, а в файл «будущее»- следующее. Массивы создаются размером 21х21.

3. Программа выдаёт пользователю на экран запрос: «Введите количество поколений». Именно столько поколений программа будет описывать.

4. Теперь программа создаёт в файле work.out поколение под номером 1. Для этого программа открывает и проверяет на наличие микробов файл (каждый символ, среди которых может затеряться микроб). Если под символом скрывается микроб (например, символ «Х»), то в массив «настоящее» записывается единица (1, т.е. микроб только что родился), а если не микроб – то ноль (0, т. е. там никто не живёт). Теперь в массиве «настоящее» находиться поле из 1 и 0 (он состоит из новорожденных младенцев и пустых мест). Все последующие поколения тоже записываются в файл work.out. Файл work.dat закрывается, и работа теперь ведётся только с файлом work.out.

5. Теперь описывается следующее поколение. Оно создаётся после проверки массива «настоящее». Проверяются микробы и их соседи (результат записывается в массив «будущее»): - если жизненный уровень микроба от 1 до 11 то: --если соседей 2 или 3, то микроб продолжает жить и подрастает (т.е. его жизненный уровень возрастает на 1), --иначе микроб погибает (=0), т.к. он задыхается или умирает от скуки; - если жизненный уровень микроба 0, то микроб рождается заново (т.е. его жизненный уровень равен 1); - если жизненный уровень микроба равен 12, то микроб погибает (от старости).

6. После проверки подсчитывается количество жизней данного поколения (т.е. сколько единиц). Затем программа проверяет, есть ли кто «живой на поле» или все погибли (т.е. нули).

7. Теперь программа производит замену поколений: массив «будущее» становится «настоящим». 100

8. Программа продолжает работу с пункта 5 до тех пор, пока количество поколений (которые описывает программа) не станет равным введенным пользователем в пункте 3 или пока не погибнут все микробы (т.е. везде одни нули).

БЛОК СХЕМА?

#include <iostream>

#include <fstream>

#include <regex>

#include <map>

#include <array>

using namespace std;

char rand\_char() { return (97 + rand() % 5); }

int cell\_int(char cell) { return (int)cell - 48; }

int main() {

srand((unsigned)time(NULL));

ofstream work\_dat("work.dat");

fstream work\_out("work.out");

for (int i = 0; i < 21; i++) {

for (int j = 0; j < 21; j++) {

char ch = rand\_char();

work\_dat << ch << " ";

}

work\_dat << endl;

}

work\_dat.close();

array<array<int, 21>, 21> present;

array<array<int, 21>, 21> future;

cout << "Enter number of generations: ";

int steps; cin >> steps;

if (!cin) return 1;

char symbol = rand\_char();

ifstream fin("work.dat");

cout << endl << "Generation 0" << endl;

int c = 0;

for (int i = 0; i < 21; i++) {

char ch;

for (int j = 0; j < 21 && fin >> ch; j++) {

char cell = (ch == symbol ? '1' : 'x');

work\_out << cell << " ";

cout << cell << " ";

present[i][j] = cell;

c += cell != 'x';

}

cout << endl;

work\_out << endl;

}

fin.close();

cout << "Living cells: " << c << endl;

for (int i = 0; i < steps && c != 0; i++) {

cout << endl << "Generation " << i + 1;

c = 0;

cout << endl;

work\_out.clear();

for (int i = 0; i < 21; i++) {

for (int j = 0; j < 21; j++) {

char cell = present[i][j];

if (cell == 'x') {

cout << cell;

}

else {

if (cell\_int(cell) > 0 && cell\_int(cell) < 11) {

int n = 0;

for (int k = max(i - 1, 0); k <= min(i + 1, 20); k++) {

for (int l = max(j - 1, 0); l <= min(j + 1, 20); l++) {

int t = cell\_int(present[k][l]);

if (!(k == i && l == j) && present[k][l] != 'x' && t >= 1 && t <= 11) n++;

}

}

if (n == 2 || n == 3) {

cell++;

c++;

}

else cell = '0';

}

else if (cell\_int(cell) == 0) {

cell++;

c++;

}

else if (cell\_int(cell) == 11) cell = '0';

cout << cell\_int(cell);

}

cout << " ";

future[i][j] = cell;

work\_out << cell << " ";

}

cout << endl;

work\_out << endl;

}

cout << "Living cells: " << c << endl;

present = future;

}

if (c == 0) cout << endl << "Oh no! Everyone died!" << endl;

work\_out.close();

}

Эта строчка нужна, чтобы функция rand() возвращалв

13 srand((unsigned)time(NULL));