Основы программирования Лабораторная работа №9

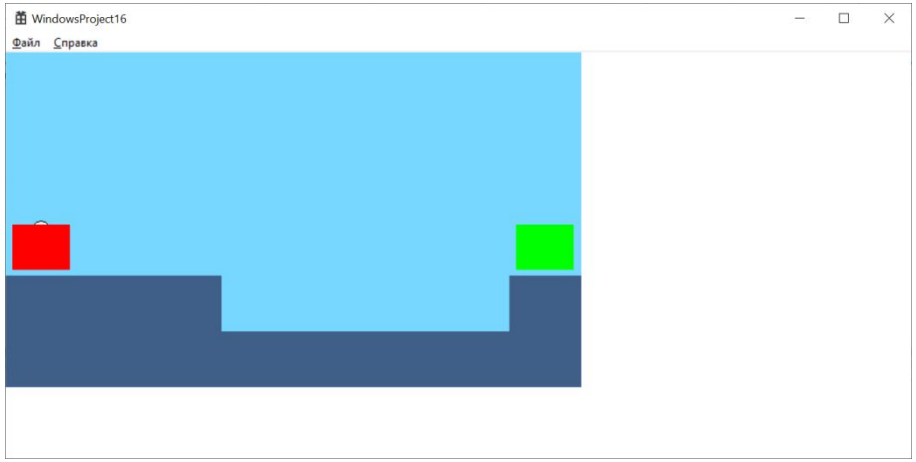
Структуры
Работа с файлами.
Текстовое меню.
(и это всё с FOR!)

Власенко О.Ф.

Задача сквозная

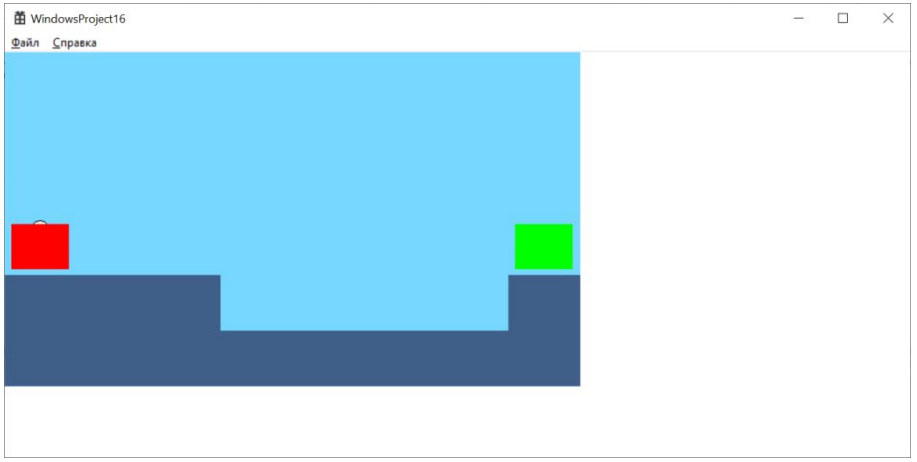
Делаем логику для игры.
В игре есть вертикальная карта – разрез земли (смотри SuperMario и подобные игры).
На этой карте есть элементы двух типов – «воздух» = 0 и «земля» = 1
Есть точка входа в карту – на картинке это красный прямоугольник.
И точка выхода – зеленый прямоугольник.

Ниже приведено отображение возможного состояния игры.



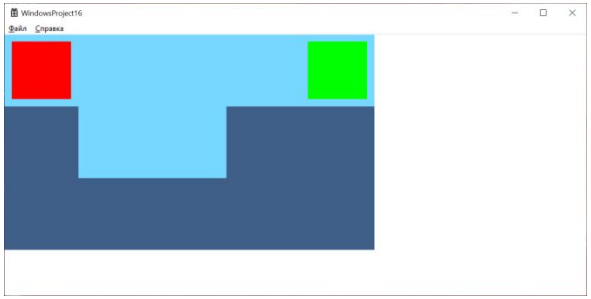
Структура для хранения игры

```
// индексы входа и выхода
struct Position {
	int i, j;
};

// Уровень игры
struct Level {
	int map[10][10]; // карта уровня
// 0 – воздух
// 1 – земля
	int n; // количество строк
	int m; // количество столбцов
	struct Position entry; // вход
	struct Position exit; // выход
};
```



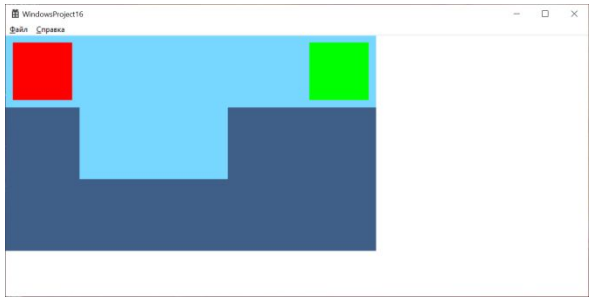
Инициализация структуры

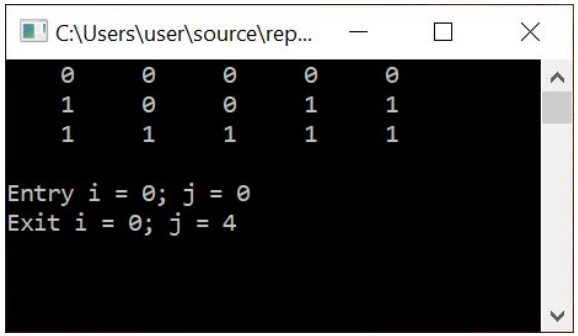
```
void main()
{
	struct Level g = {
		{
			{0, 0, 0, 0, 0},
			{1, 0, 0, 1, 1},
			{1, 1, 1, 1, 1}
		},
		3,
		5,
		{0, 0},
		{0, 4}
	};

	…
}
```



Вывод состояния игры

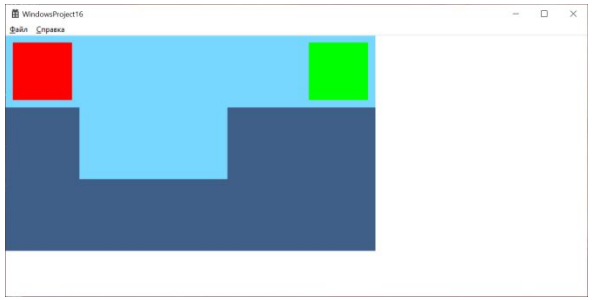
```
void main()
{
	struct Level g = {
		{
			{0, 0, 0, 0, 0},
			{1, 0, 0, 1, 1},
			{1, 1, 1, 1, 1}
		},
		3,
		5,
		{0, 0},
		{0, 4}
	};

	printLevel(&g);
	{
		int x;
		scanf("%d", &x);
	}
}
```

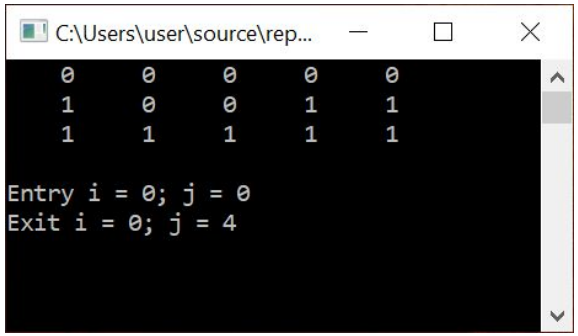




Вывод состояния игры (2)

```
// вывод массива в консоль
void printLevel(struct Level  * level) {

	for (int i = 0; i < level->n; i++) {
		for (int j = 0; j < level->m; j++) {
			printf("%5d ", level->map[i][j]);
		}
		printf("\n");
	}
	printf("\n");

	printf("Entry i = %d; j = %d\n", level->entry.i, level->entry.j);
	printf("Exit i = %d; j = %d\n\n", level->exit.i, level->exit.j);
}
```



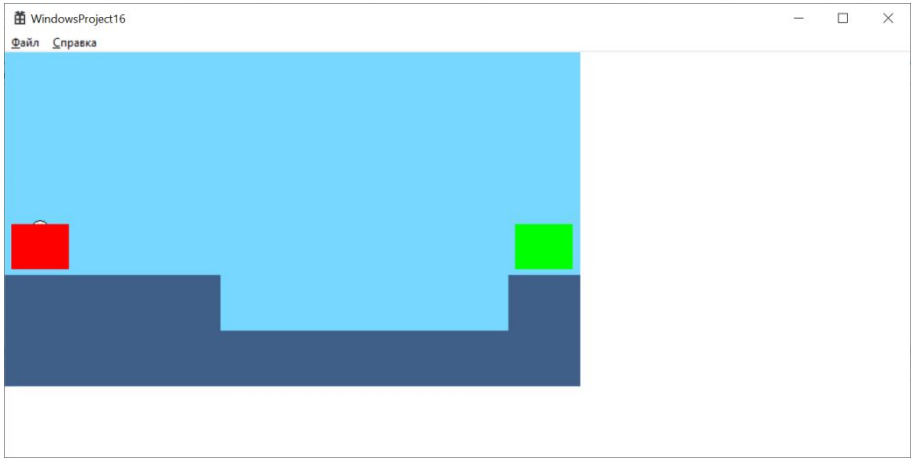


Файл с состоянием игры - формат файла

Формат файла с состоянием игры:

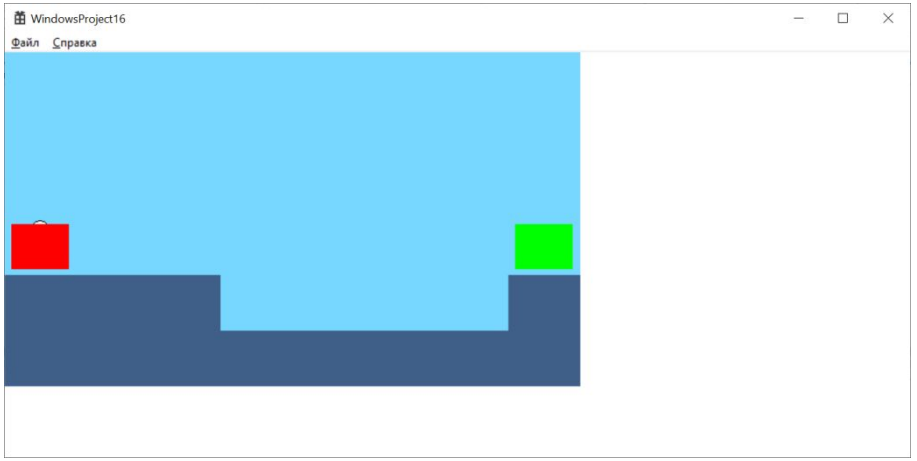
N M
N строк по M элементов в каждой
I_входа J_входа
I_выхода J_выхода

Пример файла – смотри на следующем слайде.



Файл с состоянием игры - пример

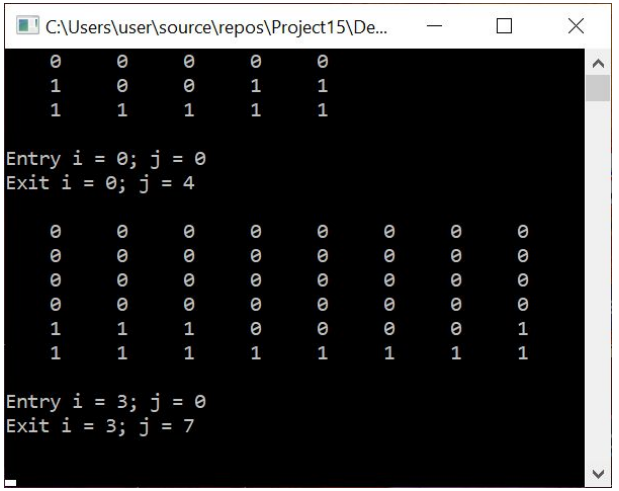
Ниже приведено содержимое файла, который хранит состояние, приведенное на картинке справа.




```
6 8
0 0 0 0 0 0 0 0
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 1
1 1 1 1 1 1 1 1
3 0
3 7
```

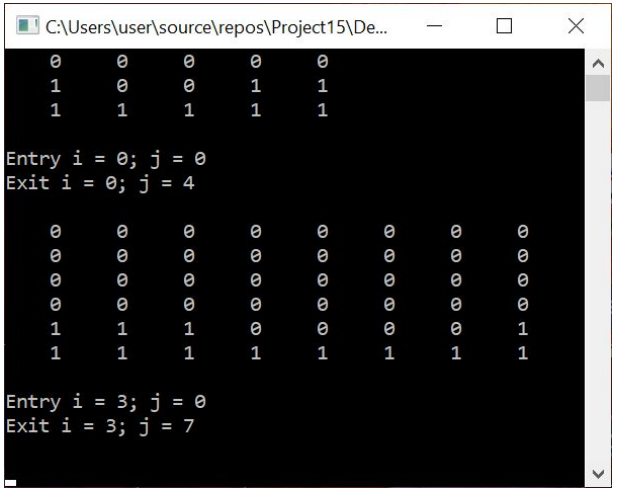
Загрузка состояния игры – вызов функции

```
void main() {
    struct Level g = {
        {   {0, 0, 0, 0, 0},
            {1, 0, 0, 1, 1},
            {1, 1, 1, 1, 1}
        },
        3,
        5,
        {0, 0},
        {0, 4}
    };
    printLevel(&g);

    loadLevel(&g);
    printLevel(&g);
    {
        int x;
        scanf("%d", &x);
    }
}
```



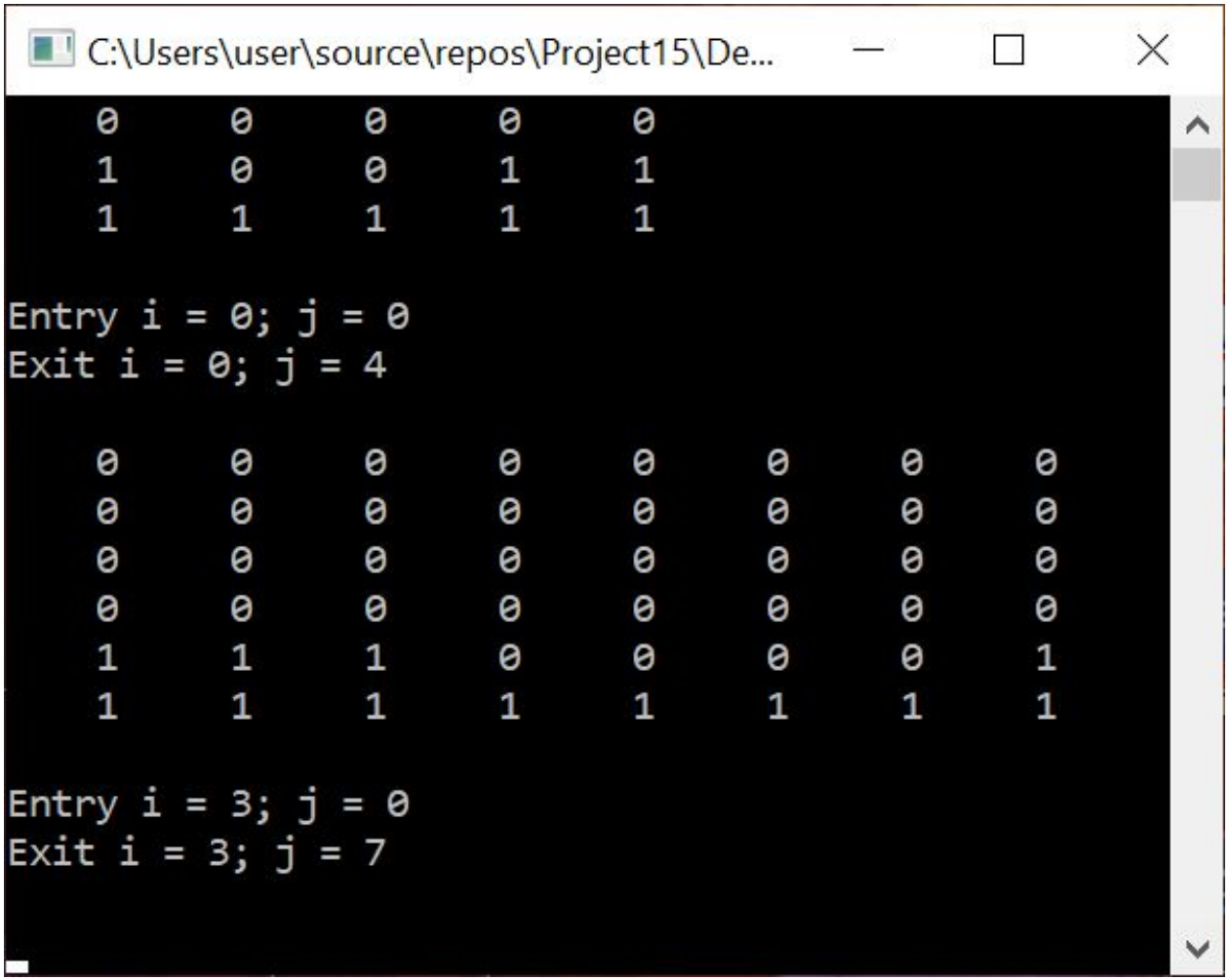
Загрузка состояния игры – функция!

```
char filename[] = "d:\\Temp\\Files\\Lab9\\1.txt";

int loadLevel(struct Level  * level) {
	FILE *fin = fopen(filename, "rt");

	if (fin == NULL) {
		printf("File %s is not opened", filename);
		return 0;
	}

	fscanf(fin, "%d", &level->n);
	fscanf(fin, "%d", &level->m);

	for (int i = 0; i < level->n; i++) {
		for (int j = 0; j < level->m; j++) {
			fscanf(fin, "%d", &level->map[i][j]);
		}
	}

	fscanf(fin, "%d", &level->entry.i);
	fscanf(fin, "%d", &level->entry.j);

	fscanf(fin, "%d", &level->exit.i);
	fscanf(fin, "%d", &level->exit.j);

	fclose(fin);
	return 1;
}
```



Задача 1 – собрать весь код!

Из кусков кода выше – соберите работающую программу, которая выдает на экран (см скриншот!)

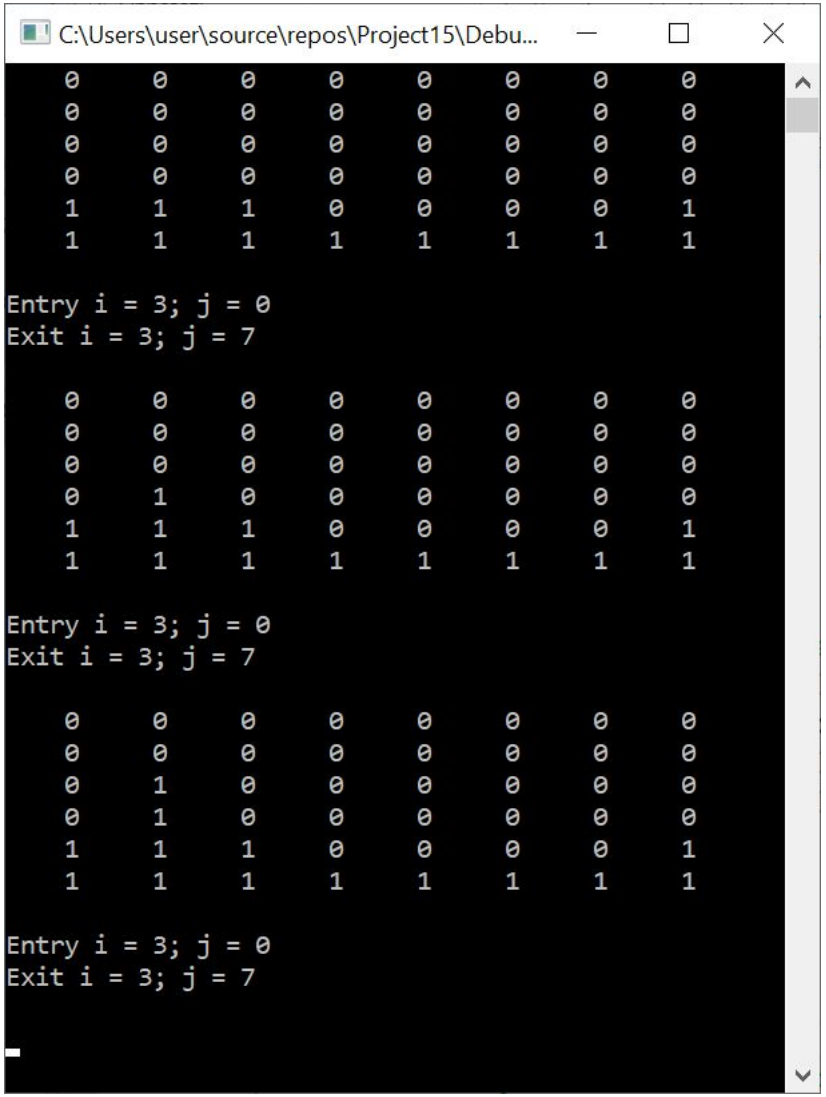


Задача 2 – Добавить элемент земли в заданный столбец (функция)

```c
// Добавление элемента "Земля" сверху в заданный столбец
void addInColumn(struct Level  * level, int indexCol) {
    for (int i = level->n - 1; i >= 0; i--) {
        if (level->map[i][indexCol] == 0) {
            level->map[i][indexCol] = 1;
            break;
        }
    }
}
```

Задача 2 – Добавить элемент земли в заданный столбец (вызов)

```
void main()
{
    struct Level g;

    loadLevel(&g);
    printLevel(&g);

    addInColumn(&g, 1);
    printLevel(&g);

    addInColumn(&g, 1);
    printLevel(&g);
    {
        int x;
        scanf("%d", &x);
    }
}
```



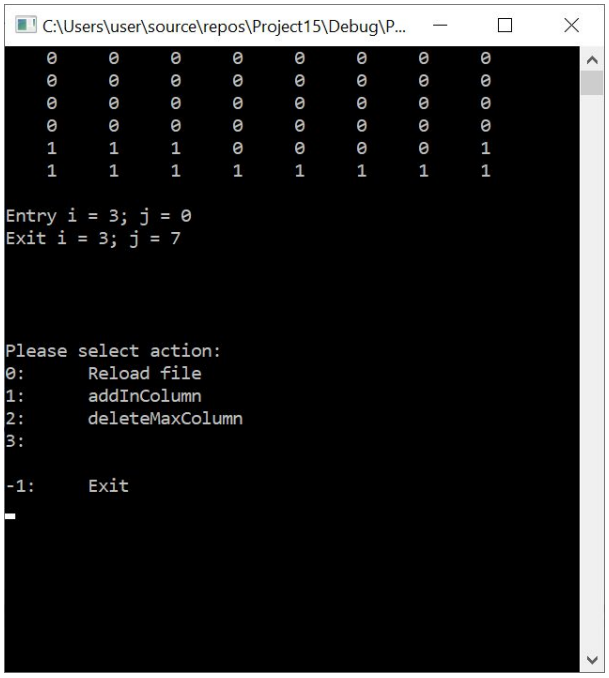
Задача 3 – Сделать меню для выбора действий (1)

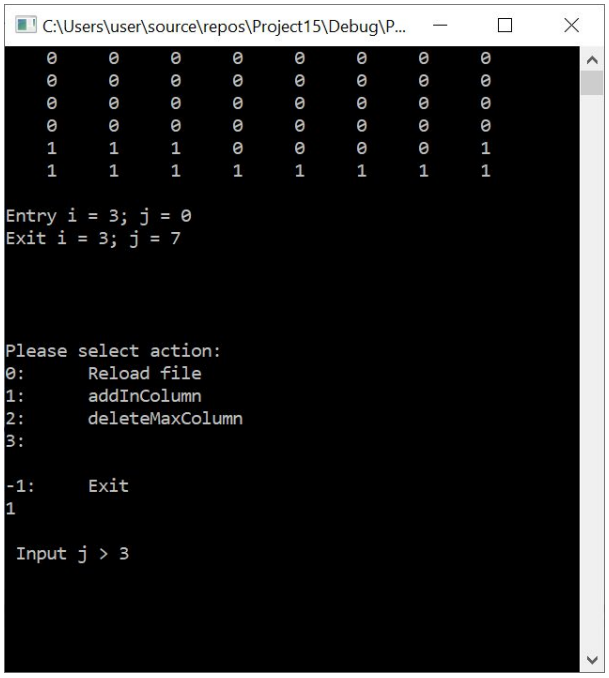
```
void main() {
    struct Level g;
    int k = 0;
    do
    {
        if (k == 0) {
            loadLevel(&g);
        }
        if (k == 1) {
            int j;
            printf("\n Input j > ");
            scanf("%d", &j);
            addInColumn(&g, j);
        }
        if (k == 2) {
            //deleteMaxColumn(&g);
        }
        if (k == 3) {
            // здесь нужно вставить вызов нового метода
        }
```

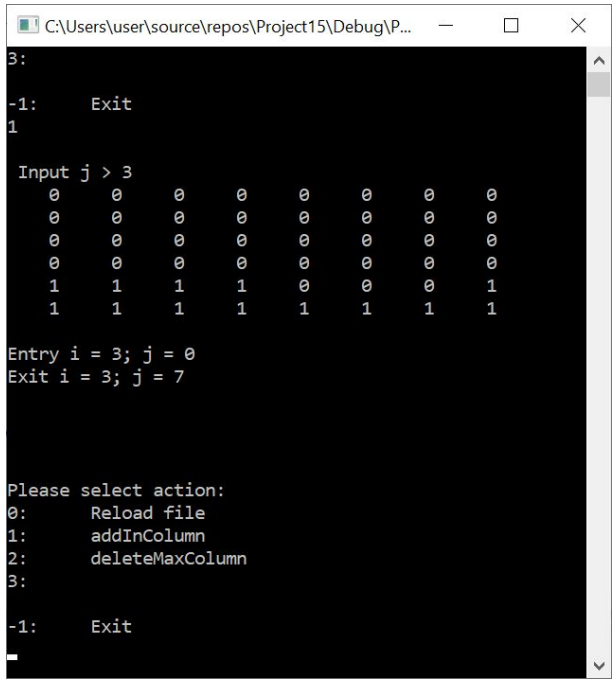
Задача 3 – Сделать меню для выбора действий (2)

```
        printLevel(&g);

        printf("\n\n\n");
        printf("Please select action:\n");
        printf("0:  Reload file\n");
        printf("1:  addInColumn\n");
        printf("2:  deleteMaxColumn\n");
        printf("3:  \n");
        printf("\n-1:   Exit\n");

        // Добавление элементов  в самую низкую горку
        // Удаление элементов из заданной горки
        // Сохранение состояния уровня
        scanf("%d", &k);
    } while (k != -1);
}
```

Задача 3 – Сделать меню для выбора действий (3)







Задача 4 – удалить «землю» из заданной горки

Создайте функцию, которая будет удалять верхний элемент «земля» из заданной высокой горки (т.е. из столбца, индекс которого задан)

Задача 5 – срыть самую высокую горку

Создайте функцию, которая будет удалять верхний элемент «земля» из самой высокой горки (т.е. из столбца, где больше всего «земли»)

Задача 6 – подсыпать земли в низину

Создайте функцию, которая будет добавлять сверху элемент «земля» в самую низкую часть(т.е. в столбец, где меньше всего «земли»)

Задача 7 – Сохранить состояние игры в файле

Создайте функцию, которая будет сохранять в файл состояние игры – в формате, аналогичном входному файлу.

Домашнее задание

Домашняя работа по лабораторной работе №9 включает в себя

(Делается по материалам классной работы)

1) Создать структуру для хранения двухмерного массива
2) Реализовать загрузку массива из файла в структуру
3) Реализовать необходимый набор действий
4) Реализовать сохранения массива из структуры в файл – структура выходного файла аналогична структуре входного
5) Подготовить отчет (со стандартным содержанием - титульный лист, задание, распечатка, блоксхемы методов)

ОБЯЗАТЕЛЬНО использовать в домашней работе:

1) Структуры для хранения массива
2) Файлы для входа и выхода
3) Цикл FOR

Домашнее задание - варианты

Вариант 1:
В массиве все элементы, стоящие выше максимального элемента, заменить на максимальный элемент первого столбца.

Вариант 2:
В массиве все элементы, стоящие выше максимального элемента, заменить на минимальный элемент последней строки.

Вариант 3:
В массиве все элементы, стоящие выше и левее минимального элемента, заменить на среднее арифметическое минимального и максимального элементов.

Вариант 4:
В массиве все элементы, стоящие ниже и левее максимального элемента, заменить на среднее арифметическое минимального и максимального элементов последнего столбца.

Вариант 5:
В массиве все элементы, стоящие ниже и левее максимального элемента, заменить на минимальный элемент.

Вариант 6:
В массиве все нечетные элементы, стоящие ниже минимального элемента массива и стоящие слева от максимального элемента массива, заменить на 0.

Вариант 7:
В массиве все четные элементы, стоящие снизу от максимального элемента массива, заменить на максимальный элемент столбца, в котором они расположены.

Вариант 8:
В массиве все нечетные элементы, стоящие сверху от минимального элемента массива, заменить на максимальный элемент строки, в которой они расположены.

Вариант 9:
В массиве все элементы, имеющие четное значение суммы индексов, заменить на минимальный элемент массива.

Вариант 10:
Обнулить элементы в тех столбцах, в которых встречается хотя бы два одинаковых элемента.

Альтернативные варианты – смотри

Домашнее задание – альтернативные варианты

Выберите себе игру, которую вы хотели бы реализовать.
Игра должна быть такой, чтобы наилучшая ее реализация была на основе двухмерного массива.

Реализуйте основной алгоритм из этой игры.

Вдохновение можно получить изучая этот документ:
https://docs.google.com/document/d/1tHW6VXBzvQb8nYH_AVJS3KDsoXh00K7GO2ZF6LZeBS8/edit?usp=sharing