

Министерство образования и науки РФ  
Федеральное государственное автономное образовательное учреждение высшего  
образования  
«Омский государственный технический университет»

Факультет (институт) Информационных технологий и компьютерных систем

Кафедра Прикладная математика и фундаментальная информатика

## Расчетно-графическая работа

по дисциплине Дискретная математика

на тему Разработка программы «Нахождения минимального пути»

Пояснительная записка

Шифр проекта 020-РГР-02.03.02-№ 20-ПЗ

Студента Остапченко Евгения Антоновича

фамилия, имя, отчество полностью

Курс 1

Группа ФИТ-221

Направление (специальность) 02.03.02

Фундаментальная информатика и информационные технологии

код, наименование

Руководитель ст. преподаватель

ученая степень, звание

Федотова И.В.

фамилия, инициалы

Выполнил \_\_\_\_\_

дата, подпись студента

**Работа защищена с количеством баллов**

\_\_\_\_\_  
дата, подпись руководителя

Омск 2023

## Содержание

Задание.....	3
Математическая формулировка задачи .....	4
Текст программы на языке C#.....	5
Список литературы .....	12

## **Задание**

Произошло радиоактивное заражение местности. Составлена карта зараженности. Она представляет собой матрицу  $N \times M$ , в клетках которой записана зараженность соответствующего участка. Петя находится в левом верхнем поле заданной матрицы.

Необходимо написать программу, которая найдет путь из левой верхней клетки таблицы в правую нижнюю клетку с минимальной суммарной дозой радиации.

### **Формат входных данных**

Входной файл содержит в первой строке числа  $N$  и  $M$ , а в следующих  $N$  строках – по  $M$  чисел – карта зараженности местности. Числа в строках разделяются одним пробелом.  $1 \leq N \leq 30$ ,  $1 \leq M \leq 30$ , зараженность участка – целое число от 0 до 100.

### **Формат выходных данных**

Выходной файл должен содержать одно число – суммарную дозу радиации.

## Математическая формулировка задачи

Задание аналогично 18 заданию из ЕГЭ по информатике.  
Воспользуемся следующим алгоритмом:

1) В новую таблицу перепишем элемент  $a_{11}$ . Далее заполним первую строку (начиная с  $a_{12}$ ) таблицы суммой предыдущего элемента и нынешнего, то есть по формуле:

$$a_{1j} = a_{1(j-1)} + b_{1j}, \text{ где } b - \text{элементы начальной таблицы, } j = 1 \dots M.$$

2) Аналогично делаем для первого столбца, по формуле:

$$a_{i1} = a_{(i-1)1} + b_{i1}, \text{ где } b - \text{элементы начальной таблицы, } i = 1 \dots N.$$

3) Теперь заполним середину таблицы. Поскольку нам нужен минимальный путь, то мы выбираем минимум среди элементов, которые расположены слева и сверху относительно искомого, не забываем добавить сам элемент. Получаем формулу:

$$a_{ij} = \min(a_{(i-1)j}; a_{i(j-1)}) + b_{ij}, \text{ где } b - \text{элементы начальной таблицы, } j = 1 \dots M, i = 1 \dots N.$$

4) Последний получившийся элемент таблицы (правая нижняя клетка) и есть путь с минимальной суммой.

## Текст программы на языке C#

```
static int[,] Work(int[,] nmas, int[,] mas){
    var n=nmas.GetUpperBound(0)+1;
    var m=nmas.GetUpperBound(1)+1;

    for (int i=1;i<n;i++){
        for(int j=1;j<m;j++){
            var verh=nmas[i-1,j];
            var bok=nmas[i,j-1];

            if (verh<bok) nmas[i,j]=mas[i,j]+verh;

            if (bok<verh) nmas[i,j]=mas[i,j]+bok;

            if (verh==bok) nmas[i,j]=mas[i,j]+bok;

        }
    }
    return nmas;
}

static int[,] RoundWork(int[,] mas){
    var n=mas.GetUpperBound(0)+1;
    var m=mas.GetUpperBound(1)+1;
    int[,] nmas=new int[n,m];
    for (int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            if (i==0 && j==0) nmas[i,j]=mas[i,j];

            if (i==0 && j!=0) nmas[i,j]=mas[i,j]+nmas[i,j-1];

            if (i!=0 && j==0) nmas[i,j]=mas[i,j]+nmas[i-1,j];

        }
    }
    return nmas;
}

static void Print(int[,] prmas){
    var n=prmas.GetUpperBound(0)+1;
    var m=prmas.GetUpperBound(1)+1;
    //Console.WriteLine(new string('-',m*7));
    for (int i=0;i<n;i++){
        Console.WriteLine(new string('-',m*6));
        Console.Write("| ");
        for(int j=0;j<m;j++){
            Console.Write( "{0,4} |", prmas[i,j]);
        }
    }
}
```

```

        if(j==m-1) Console.WriteLine();
    }
};
Console.WriteLine(new string('-',6*m));
}

static void Avtor()
{
    Console.WriteLine();
    Console.WriteLine("Выполнил: Остапченко Евгений");
    Console.WriteLine("Курс: 1");
    Console.WriteLine("Группа: ФИТ-221");
    //Menu();
}

static int[,] Zapolnenie_RND(){
    Random rnd=new Random();

    var n=rnd.Next(1,30);
    var m=rnd.Next(1,30);
    int[,] mas=new int[n,m];
    for (int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            var num=rnd.Next(0,100);
            mas[i,j]=num;
            //Console.WriteLine();
        }
    };
    return mas;
}

static int[,] Zapolnenie(int n,int m){
    int[,] mas=new int[n,m];
    for (int i=0;i<n;i++){
        for(int j=0;j<m;j++){
            Console.Write( " Введите элемент {0},{1}: ", i,j);
            int num=Convert.ToInt32(Console.ReadLine());
            mas[i,j]=num;
            //Console.WriteLine();
        }
    };
    return mas;
}

static void Menu()
{

```

```

Console.WriteLine("\nМеню РГР:");
Console.WriteLine("1. Ручное заполнение");
Console.WriteLine("2. Рандомное заполнение");
Console.WriteLine("3. Об авторе");
Console.WriteLine("4. Выход\n");
Console.Write("Введите действие: ");
int nom=Convert.ToInt32(Console.ReadLine());
switch (nom){
    case 1:
        Console.WriteLine("Количество строк N: ");
        var n=Convert.ToInt32(Console.ReadLine());
        Console.WriteLine("Количество столбцов M: ");
        var m=Convert.ToInt32(Console.ReadLine());

        var mas=Zapolnenie(n,m);
        var rez_mas=RoundWork(mas);
        rez_mas=Work(rez_mas, mas);
        Console.WriteLine("\nКарта:");
        Print(mas);
        Console.WriteLine("\nЗаражение:");
        Print(rez_mas);
        Console.WriteLine("Суммарное заражение: {0}",rez_mas[n-1,m-1]);
        Console.WriteLine("\nНажмите Enter для продолжения...");
        if(Console.ReadKey()!=null) Menu();
        break;

    case 2:
        var mas_rnd=Zapolnenie_RND();
        var rnd_mas=RoundWork(mas_rnd);
        var rez_mas_rnd=Work(rnd_mas, mas_rnd);
        Console.WriteLine("\nКарта:");
        Print(mas_rnd);
        Console.WriteLine("\nЗаражение:");
        Print(rez_mas_rnd);
        Console.WriteLine("Суммарное заражение:
{0}",rez_mas_rnd[mas_rnd.GetUpperBound(0),mas_rnd.GetUpperBound(1)]);

        Console.WriteLine("\nНажмите Enter для продолжения...");
        if(Console.ReadKey()!=null) Menu();
        break;
    case 3:
        Avtor();

        Console.WriteLine("\nНажмите Enter для продолжения...");
        if(Console.ReadKey()!=null) Menu();
        break;
    case 4:
        Console.WriteLine("\nДосвидание!");
        break;
    default:

```

```
        Console.WriteLine("Ошибка!");

        Console.WriteLine("\nНажмите Enter для продолжения...");
        if(Console.ReadKey() != null) Menu();
        break;
    }
}

Menu();
```



## Разработка интерфейса пользователя

На рисунке 2 представлено основное меню программы. Управление осуществляется при помощи ввода номера пункта меню с клавиатуры и нажатия клавиши Enter.

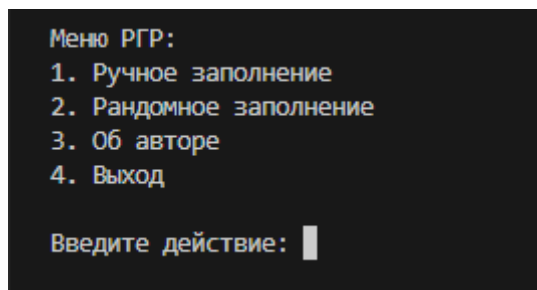


Рисунок 2 – Основное меню программы

На рисунке 3 представлена функция ручного заполнения таблицы и поиска минимального пути.

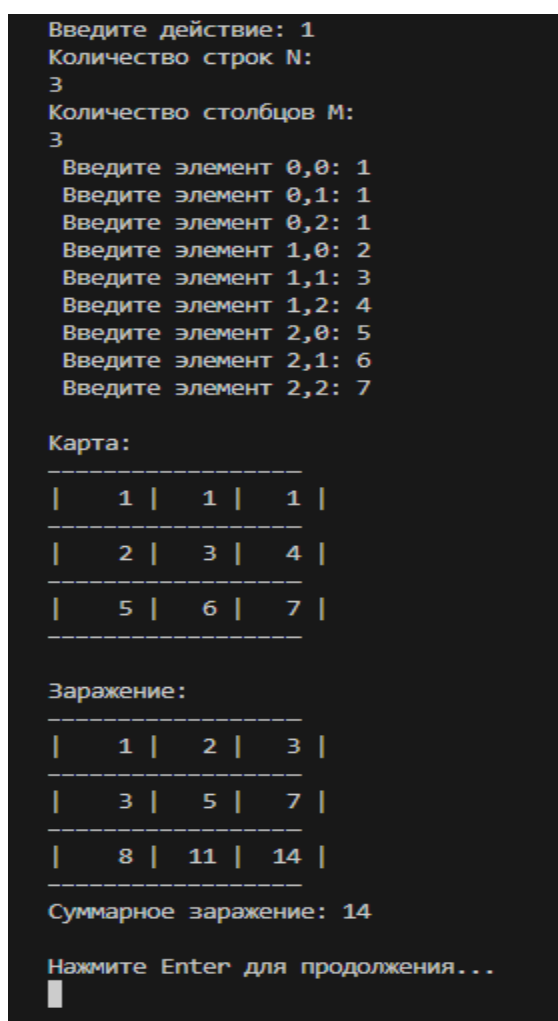


Рисунок 3 – Функция ручного заполнения таблицы для поиска минимального пути

На рисунках 4 и 5 представлена функция случайного заполнения таблицы и поиска минимального пути.

Введите действие: 2

Карта:

	51		61		51		56		6		34		10		61		67		72		5		24		68		31		32		13		8		69		38		3		16		44	
	24		31		45		15		90		43		1		3		83		3		68		26		16		12		76		51		16		80		74		37		60		51	
	25		74		72		14		71		31		65		67		63		68		98		86		16		8		54		69		37		9		10		88		7		70	
	94		0		97		98		36		45		83		82		28		67		23		92		28		26		94		9		79		29		39		28		60		74	
	72		64		83		36		67		59		97		88		98		91		42		36		90		37		99		75		61		42		64		52		84		37	
	78		65		37		14		56		20		37		51		75		58		8		30		6		49		38		93		19		14		51		84		16		26	
	88		88		92		67		27		22		13		48		51		20		55		69		73		71		40		48		13		57		81		67		37		86	
	88		50		52		77		29		43		84		1		67		92		82		45		62		15		93		12		15		4		75		59		31		70	
	22		24		49		3		60		75		66		90		46		84		59		64		59		10		89		53		64		34		51		25		5		36	
	12		34		55		24		19		78		96		61		47		89		3		52		46		40		79		24		18		67		13		89		66		94	
	53		30		74		43		26		63		12		66		78		12		50		63		4		25		74		14		34		68		23		93		77		53	
	10		11		18		77		26		41		71		98		84		95		94		73		57		97		15		67		76		86		7		81		66		1	
	71		82		58		59		20		70		50		14		71		19		57		69		71		72		31		32		38		50		48		70		7		88	
	20		42		25		3		4		35		64		91		84		33		31		82		90		2		48		1		70		82		80		23		32		1	
	51		1		69		41		85		51		24		47		55		89		70		68		79		66		91		0		88		42		65		27		31		92	
	88		70		91		16		50		29		76		42		8		58		63		75		53		17		63		73		56		47		95		17		68		53	
	58		2		83		5		48		48		96		99		15		73		89		73		67		81		69		92		14		19		37		15		5		63	
	27		56		66		14		36		4		87		31		79		36		86		34		75		29		71		91		50		23		4		0		48		65	
	92		11		62		73		25		10		69		83		39		14		82		46		75		44		37		1		52		78		57		65		46		78	

Рисунок 4 – Функция случайного заполнения таблицы.

Заражение:

	51		112		163		219		225		259		269		330		397		469		474		498		566		597		629		642		650		719		757		760		776		820	
	75		106		151		166		256		299		270		273		356		359		427		453		469		481		557		608		624		704		778		797		836		871	
	100		174		223		180		251		282		335		340		403		427		525		539		485		489		543		612		649		658		668		756		763		833	
	194		174		271		278		287		327		410		422		431		494		517		609		513		515		609		618		697		687		707		735		795		869	
	266		238		321		314		354		386		483		510		529		585		559		595		603		552		651		693		754		729		771		787		871		906	
	344		303		340		328		384		404		441		492		567		625		567		597		603		601		639		732		751		743		794		871		887		913	
	432		391		432		395		411		426		439		487		538		558		613		666		676		672		679		727		740		797		875		938		924		999	
	520		441		484		472		440		469		523		488		555		647		695		711		738		687		772		739		754		758		833		892		923		993	
	542		465		514		475		500		544		589		578		601		685		744		775		797		697		786		792		818		792		843		868		873		909	
	554		499		554		499		518		596		685		639		648		737		740		792		838		737		816		816		834		859		856		945		939		1003	
	607		529		603		542		544		607		619		685		726		738		788		851		842		762		836		830		864		927		879		972		1016		1056	
	617		540		558		619		570		611		682		780		810		833		882		924		899		859		851		897		940		1013		886		967		1033		1034	
	688		622		616		675		590		660		710		724		795		814		871		940		970		931		882		914		952		1002		934		1004		1011		1099	
	708		664		641		644		594		629		693		784		868		847		878		960		1050		933		930		915		985		1067		1014		1027		1043		1044	
	759		665		710		685		679		680		704		751		806		895		948		1016		1095		999		1021		915		1003		1045		1079		1054		1074		1136	
	847		735		801		701		729		709		780		793		801		859		922		997		1050		1016		1079		988		1044		1091		1174		1071		1139		1189	
	905		737		820		706		754		757		853		892		816		889		978		1051		1117		1097		1148		1080		1058		1077		1114		1086		1091		1154	
	932		793		859		720		756		760		847		878		895		925		1011		1045		1120		1126		1197		1171		1108		1100		1104		1086		1134		1199	
	1024		804		866		793		781		770		839		922		934		939		1021		1067		1142		1170		1207		1172		1160		1178		1161		1151		1180		1258	

Суммарное заражение: 1258

Нажмите Enter для продолжения...

Рисунок 5 – Функция вычисления минимального пути.

На рисунке 6 представлена подпрограмма «Об авторе».

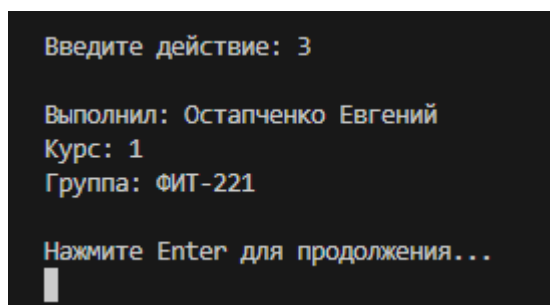


Рисунок 6 – Подпрограмма «Об авторе».

## **Список литературы**

- 1) Программирование на С# для начинающих. Основные сведения  
- Алексей Васильев 2018 г.
- 2) [https://code\\_enjoy.ru/ege\\_po\\_informatike\\_2022\\_zadanie\\_18\\_tablica\\_chisel/](https://code_enjoy.ru/ege_po_informatike_2022_zadanie_18_tablica_chisel/) (Дата обращения 15.05.2023)