

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ  
РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«ДАГЕСТАНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ»

## **РАБОЧАЯ ПРОГРАММА ДИСЦИПЛИНЫ**

**«Производственная практика, научно-исследовательская  
работа»**

Кафедра дифференциальных уравнений и функционального анализа  
факультета математики и компьютерных наук

Образовательная программа бакалавриата  
01.03.01 Математика

Направленность (профиль) программы:  
Математика

Форма обучения  
очная

Статус дисциплины: входит в обязательную часть ОПОП

Рабочая программа дисциплины «Производственная практика, научно-исследовательская работа») составлена в 2025 году в соответствии с требованиями ФГОС ВО бакалавриата по направлению подготовки 01.03.01 Математика от 10.01.2018 № 8

Разработчик: кафедра дифференциальных уравнений и функционального анализа, Ибрагимов Мурад Гаджиевич, к. ф.-м. н., доцент.

Рабочая программа дисциплины одобрена:  
на заседании кафедры дифференциальных уравнений и функционального анализа от «21» января 2025 г., протокол № 5.

Зав. кафедрой  Сиражудинов М.М.

на заседании Методической комиссии факультета математики и компьютерных наук от «23» января 2025 г., протокол № 3.

Председатель  Ризаев М.К.

Рабочая программа дисциплины согласована с учебно-методическим управлением «30» января 2025 г.

Начальник УМУ  Саидов А.Г.

**1. ПАСПОРТ  
ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ  
по дисциплине**

“учебной практике, научно-исследовательской работе (получение первичных навыков научно исследовательской работы) ”

**1.1. Основные сведения о дисциплине**

Общая трудоемкость дисциплины составляет 3 зачетные единицы (108 академических часов).

Вид работы	Трудоемкость, академических часов		
	2 семестр	4 семестр	всего
<b>Общая трудоёмкость</b>	108	108	216
<b>Контактная работа:</b>	54	54	108
Лекции (Л)			
Лабораторные занятия (ЛЗ)			
Консультации			
Промежуточная аттестация (зачет, экзамен)	зачет	зачет	
<b>Самостоятельная работа</b>	54	54	108
1. работа с лекционным материалом, с учебной литературой	10	10	20
2. опережающая самостоятельная работа (изучение нового материала до его изложения на занятиях)	10	10	20
3. выполнение домашних заданий, домашних контрольных работ	10	10	20
4. подготовка к лабораторным работам	10	10	20
5. подготовка к контрольным работам, коллоквиумам	14	14	28
6. подготовка к зачету			

**1.2. Требования к результатам обучения по дисциплине, формы их контроля и виды оценочных средств**

*ПАСПОРТ ФОНДА ОЦЕНОЧНЫХ СРЕДСТВ*

по “учебной практике, научно-исследовательской работе (получение первичных навыков научно исследовательской работы) ”»

№п/п	Контролируемые модули, разделы (темы) дисциплины	Код контролируемой компетенции (или её части)	Оценочные средства		Способ контроля
			наименование	№№ заданий	
1	Разработка графических приложений. Программирование на основе классов. Объектно-	ПК-1 ПК-2 ПК-3 ПК-4 ОПК-3 ОПК-4 УК-1	Задания для самостоятельного решения	1-3	Письменно

	ориентированно е программирован ие	УК-3			
		ПК-1 ПК-2 ПК-3 ПК-4 ОПК-3 ОПК-4	Задания для самостоятельно го решения	1-28	письменно
		ПК-1 ПК-2 ПК-3 ПК-4 ОПК-3 ОПК-4	Задания для самостоятельно го решения	1-2	письменно
2	Разработка графических приложений. Программирован ие на основе классов. Объектно- ориентированное программирован ие	ПК-1 ПК-2 ПК-3 ПК-4	Задания для самостоятельн ого решения	1-27	Письме нно
		ПК-1 ПК-2 ПК-3 ПК-4	Задания для самостояте льного решения	20-60	письмен но
		ПК-1 ПК-2 ПК-3 ПК-4	Задания для самостоятель ного решения	15-30	письмен но

### 1.3. Показатели и критерии определения уровня сформированности компетенций

№ п/п	Код компе т енции	Уровни сформированности компетенции			
		Недостаточный	Удовлетворитель ный (достаточный)	Базовый	Повышенный
	ОПК-3	Не знает на достаточном уровне основные принципы педагогической деятельности научные знания в сфере математики и информатики. Не умеет на достаточном уровне использовать современные информационноком муникационн ые	Знает на достаточном уровне основные принципы педагогической деятельности научные знания в сфере математики и информатики. программирования, технологии создания и эксплуатации программных продуктов и программных комплексов. Умеет на достаточном уровне	Знает на хорошем уровне основные принципы педагогической деятельности научные знания в сфере математики и информатики. Умеет на хорошем уровне использовать современные информационноко ммуникационн ые технологии в	Знает в совершенстве основные принципы педагогической деятельности научные знания в сфере математики и информатики. Умеет в совершенстве использовать современные информационнокомму никационн ые технологии в профессиональн ой деятельности. Владеет в совершенстве практическими навыками применять в

		технологии в профессиональной деятельности. Не владеет на достаточном уровне навыками применять в педагогической деятельности научные знания в сфере математики и информатики. практическими	использовать современные информационнокоммуникационные технологии в профессиональной деятельности. Владеет на достаточном уровне практическими навыками применять в педагогической деятельности научные знания в сфере математики и информатики.обеспечения.	профессиональной деятельности. Владеет на хорошем уровне практическими навыками применять в педагогической деятельности научные знания в сфере математики и информатики.	педагогической деятельности научные знания в сфере математики и информатики.
	ОПК-4	Не знает основные положения и концепции развития существующих информационнокоммуникационных технологий и с учетом основных требований информационной безопасности. Не умеет использовать их в профессиональной деятельности. Не имеет практические навыки разработки информационнокоммуникационных технологий с учетом основных требований информационной безопасности.	Знает на достаточном уровне основные положения и концепции развития существующих информационнокоммуникационных технологий и с учетом основных требований информационной безопасности. Умеет на достаточном уровне использовать их в профессиональной деятельности. Имеет на достаточном уровне практические навыки разработки информационнокоммуникационных технологий с учетом основных требований информационной безопасности.	Знает хорошо основные положения и концепции развития существующих информационнокоммуникационных технологий и с учетом основных требований информационной безопасности. Умеет хорошо использовать их в профессиональной деятельности. Имеет хорошие практические навыки разработки информационнокоммуникационных технологий с учетом основных требований информационной безопасности.	Знает отлично основные положения и концепции развития существующих информационнокоммуникационных технологий и с учетом основных требований информационной безопасности. Умеет отлично использовать их в профессиональной деятельности. Имеет отличные практические навыки разработки информационнокоммуникационных технологий с учетом основных требований информационной безопасности.
	ПК-1	Не знает на достаточном уровне требования к педагогической деятельности по проектированию и	Знает на достаточном уровне требования к педагогической деятельности по проектированию и	Знает на хорошем уровне требования к педагогической деятельности по проектированию и реализации	Знает в совершенстве требования к педагогической деятельности по проектированию и реализации

		<p>реализации образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования. Не умеет на достаточном уровне планировать занятия по программам обучения математике и информатике с учетом уровня подготовки и психологии аудитории. Не владеет на достаточном уровне навыками разработки и проведения уроков и индивидуальных занятий по математике и информатике.</p>	<p>реализации образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования. Умеет на достаточном уровне планировать занятия по программам обучения математике и информатике с учетом уровня подготовки и психологии аудитории. Владеет на достаточном уровне навыками проведения уроков и индивидуальных занятий по математике и информатике.</p>	<p>образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования. Умеет на хорошем уровне планировать занятия по программам обучения математике и информатике с учетом уровня подготовки и психологии аудитории. и информационных технологий для формирования выводов по соответствующим научным исследованиям Владеет на хорошем уровне навыками проведения уроков и индивидуальных занятий по математике и информатике.</p>	<p>образовательного процесса в образовательных организациях дошкольного, начального общего, основного общего, среднего общего образования. Умеет в совершенстве планировать занятия по программам обучения математике и информатике с учетом уровня подготовки и психологии аудитории. Владеет в совершенстве навыками разработки и проведения уроков и индивидуальных занятий по математике и информатике.</p>
	ПК-2	<p>Не знает на достаточном уровне требования к организационно-методическому и организационно-педагогическому обеспечению основных и дополнительных образовательных программ, а также внеклассных мероприятий. Не умеет на</p>	<p>Знает на достаточном уровне требования к организационно-методическому и организационно-педагогическому обеспечению основных и дополнительных образовательных программ, а также внеклассных мероприятий. Умеет на достаточном</p>	<p>Знает на хорошем уровне требования к организационно-методическому и организационно-педагогическому обеспечению основных и дополнительных образовательных программ, а также внеклассных мероприятий. Умеет на хорошем</p>	<p>Знает в совершенстве требования к организационно-методическому и организационно-педагогическому обеспечению основных и дополнительных образовательных программ, а также внеклассных мероприятий. Умеет в совершенстве планировать урочную деятельность и</p>

		<p>достаточном уровне планировать урочную деятельность и внеклассные мероприятия на основе существующих методик. Не владеет на достаточном уровне навыками выбора оптимальных методов и методики преподавания при планировании урока.</p>	<p>уровне планировать урочную деятельность и внеклассные мероприятия на основе существующих методик. Владеет на достаточном уровне навыками выбора оптимальных методов и методики преподавания при планировании урока.</p>	<p>уровне планировать урочную деятельность и внеклассные мероприятия на основе существующих методик. Владеет на хорошем уровне навыками выбора оптимальных методов и методики преподавания при планировании урока.</p>	<p>внеклассные мероприятия на основе существующих методик. Владеет в совершенстве навыками выбора оптимальных методов и методики преподавания при планировании урока.</p>
	ПК-3	<p>Не демонстрирует признаков удовлетворительного уровня освоения компетенции.</p>	<p>Знает основы современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. Планирует популярные лекции, экскурсии и другие виды деятельности необходимые для формирования выводов по соответствующим научным исследованиям. Проводит необходимую работу по сборанию, обработке и интерпретированию современных научных исследований, необходимых для формирования выводов по соответствующим научным исследованиям.</p>	<p>Знает хорошо основы современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. Хорошо планирует популярные лекции, экскурсии и другие виды деятельности необходимые для формирования выводов по соответствующим научным исследованиям. Хорошо проводит необходимую работу по сборанию, обработке и интерпретированию современных научных исследований, необходимых для формирования выводов по соответствующим научным</p>	<p>Знает отлично основы современных научных исследований, необходимые для формирования выводов по соответствующим научным исследованиям. Отлично планирует популярные лекции, экскурсии и другие виды деятельности необходимые для формирования выводов по соответствующим научным исследованиям. Отлично проводит необходимую работу по сборанию, обработке и интерпретированию современных научных исследований, необходимых для формирования выводов по соответствующим научным исследованиям.</p>

				исследованиям.	
	ПК-4	Не демонстрирует признаков удовлетворительного уровня освоения компетенции.	Обладает базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий. Умеет находить, формулировать и решать стандартные задачи в собственной научноисследовательской деятельности в математике и информатике. Имеет практический опыт научно-исследовательской деятельности в математике и информатике.	Обладает хорошими базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий. Умеет хорошо находить, формулировать и решать стандартные задачи в собственной научноисследовательской деятельности в математике и информатике. Имеет хороший практический опыт научно-исследовательской деятельности в математике и информатике.	Обладает отличными базовыми знаниями, полученными в области математических и (или) естественных наук, программирования и информационных технологий. Умеет отлично находить, формулировать и решать стандартные задачи в собственной научноисследовательской деятельности в математике и информатике. Имеет отличный практический опыт научно-исследовательской деятельности в математике и информатике.

**1. КОНТРОЛЬНЫЕ ЗАДАНИЯ И ИНЫЕ МАТЕРИАЛЫ ОЦЕНКИ  
знаний, умений, навыков и (или) опыта деятельности,  
характеризующие этапы формирования компетенций в процессе  
освоения учебной практики  
Варианты заданий**

**Задание 1**

**Вариант 1**

Описать класс, реализующий стек. Написать программу, использующую этот класс для моделирования T-образного сортировочного узла на железной дороге. Программа должна разделять на два направления состав, состоящий из вагонов двух типов (на каждое направление формируется состав из вагонов одного типа). Предусмотреть возможность формирования состава из файла и с клавиатуры.

**Вариант 2**

Описать класс, реализующий бинарное дерево, обладающее возможностью добавления новых элементов, удаления существующих, поиска элемента по ключу, а также

последовательного доступа ко всем элементам. Написать программу, использующую этот класс для представления англо-русского словаря. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса. Предусмотреть возможность формирования словаря из файла и с клавиатуры.

### **Вариант 3**

Построить систему классов для описания плоских геометрических фигур: круга, квадрата, прямоугольника. Предусмотреть методы для создания объектов, перемещения на плоскости, изменения размеров и вращения на заданный угол. Написать программу, демонстрирующую работу с этими классами. Программа должна содержать меню, позволяющее осуществить проверку всех методов классов.

### **Вариант 4**

Построить описание класса, содержащего информацию о почтовом адресе организации. Предусмотреть возможность отдельного изменения составных частей адреса, создания и уничтожения объектов этого класса. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 5**

Составить описание класса для представления комплексных чисел. Обеспечить выполнение операций сложения, вычитания и умножения комплексных чисел. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 6**

Составить описание класса для объектов-векторов, задаваемых координатами концов в трехмерном пространстве. Обеспечить операции сложения и вычитания векторов с получением нового вектора (суммы или разности), вычисления скалярного произведения двух векторов, длины вектора, косинуса угла между векторами. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 7**

Составить описание класса прямоугольников со сторонами, параллельными осям координат. Предусмотреть возможность перемещения прямоугольников на плоскости, изменение размеров, построение наименьшего прямоугольника, содержащего два заданных прямоугольника, и прямоугольника, являющегося общей частью (пересечением) двух прямоугольников. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 8**

Составить описание класса для определения одномерных массивов целых чисел (векторов). Предусмотреть возможность обращения к отдельному элементу массива с контролем выхода за пределы массива, возможность задания произвольных границ индексов при создании объекта, возможность выполнения операций поэлементного сложения и вычитания массивов с одинаковыми границами индексов, умножения и деления всех элементов массива на скаляр, вывода на экран элемента массива по заданному индексу,

вывода на экран всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 9**

Составить описание класса для определения одномерных массивов строк фиксированной длины. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массива, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива по заданному индексу и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 10**

Составить описание класса многочленов от одной переменной, задаваемых степенью многочлена и массивом коэффициентов. Предусмотреть методы для вычисления значения многочлена для заданного аргумента, операции сложения, вычитания и умножения многочленов с получением нового объекта-многочлена, вывод на экран описания многочлена. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 11**

Составить описание класса одномерных массивов строк, каждая строка задается длиной и указателем на выделенную для нее память. Предусмотреть возможность обращения к отдельным строкам массива по индексам, контроль выхода за пределы массивов, выполнения операций поэлементного сцепления двух массивов с образованием нового массива, слияния двух массивов с исключением повторяющихся элементов, вывод на экран элемента массива и всего массива. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 12**

Составить описание класса, обеспечивающего представление матрицы произвольного размера с возможностью изменения числа строк и столбцов, вывода на экран подматрицы любого размера и всей матрицы. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### **Вариант 13**

Написать класс для эффективной работы со строками, позволяющий форматировать и сравнивать строки, хранить в строках числовые значения и извлекать их. Для этого необходимо реализовать:

- перегруженные операции присваивания и конкатенации;
- операции сравнения и приведения типов;
- преобразование в число любого типа;
- форматный вывод строки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Вариант 14**

Описать класс «домашняя библиотека». Предусмотреть возможность работы с произвольным числом книг, поиска книги по какому-либо признаку (например, по автору или по году издания), добавления книг в библиотеку, удаления книг из нее, сортировки книг по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Вариант 15**

Описать класс «записная книжка». Предусмотреть возможность работы с произвольным числом записей, поиска записи по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Вариант 16**

Описать класс «студенческая группа». Предусмотреть возможность работы с переменным числом студентов, поиска студента по какому-либо признаку (например, по фамилии, дате рождения или номеру телефона), добавления и удаления записей, сортировки по разным полям. Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Вариант 17**

Описать класс, реализующий тип данных «вещественная матрица» и работу с ними. Класс должен реализовывать следующие операции над матрицами:

- сложение, вычитание, умножение, деление (+, -, \*, /) (умножение и деление, как на другую матрицу, так и на число);
- комбинированные операции присваивания (+=, -=, \*=, /=);
- операции сравнения на равенство/неравенство;
- операции вычисления обратной и транспонированной матрицы, операцию возведения в степень;
- методы вычисления детерминанта и нормы;
- методы, реализующие проверку типа матрицы (квадратная, диагональная, нулевая, единичная, симметрическая, верхняя треугольная, нижняя треугольная);
- операции ввода/вывода в стандартные потоки.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

#### **Вариант 18**

Описать класс «множество», позволяющий выполнять основные операции — добавление и удаление элемента, пересечение, объединение и разность множеств.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

### Вариант 19

Описать класс, реализующий стек. Написать программу, использующую этот класс для отыскания прохода по лабиринту. Лабиринт представляется в виде матрицы, состоящей из квадратов. Каждый квадрат либо открыт, либо закрыт. Вход в закрытый квадрат запрещен. Если квадрат открыт, то вход в него возможен со стороны, но не с угла. Каждый квадрат определяется его координатами в матрице. После отыскания прохода программа печатает найденный путь в виде координат квадратов.

### Вариант 20

Описать класс «предметный указатель». Каждый компонент указателя содержит слово и номера страниц, на которых это слово встречается. Количество номеров страниц, относящихся к одному слову, от одного до десяти. Предусмотреть возможность формирования указателя с клавиатуры и из файла, вывода указателя, вывода номеров страниц для заданного слова, удаления элемента из указателя.

Написать программу, демонстрирующую работу с этим классом. Программа должна содержать меню, позволяющее осуществить проверку всех методов класса.

## Задание 2

### Общая часть заданий для вариантов 1-20

Написать программу, демонстрирующую работу с объектами двух типов: T1 и T2, для чего создать систему соответствующих классов. Каждый объект должен иметь идентификатор (в виде произвольной строки символов) и одно или несколько полей для хранения состояния объекта (один класс является потомком другого).

Клиенту (функции main) должны быть доступны следующие основные операции (методы): создать объект, удалить объект, показать значение объекта и прочие дополнительные операции (зависят от варианта). Операции по созданию и удалению объектов инкапсулировать в классе Factory.

Предусмотреть меню, позволяющее продемонстрировать заданные операции. При необходимости в разрабатываемые классы добавляются дополнительные методы (например, конструктор копирования, операция присваивания и т. п.) для обеспечения надлежащего функционирования этих классов.

### Варианты 1-10.

В табл. 1.1 и 1.2 перечислены возможные типы объектов и возможные дополнительные операции над ними.

**Табл. 1.1.** Перечень типов объектов

Класс	Объект
SymbString	Символьная строка (произвольная строка символов)
BinString	Двоичная строка (изображение двоичного числа)

OctString	Восьмеричная строка (изображение восьмеричного числа)
DecString	Десятичная строка (изображение десятичного числа)
HexString	Шестнадцатеричная строка (изображение шестнадцатеричного числа)

**Табл. 1.2.** Перечень дополнительных операций (методов)

Операция(метод)	Описание
ShowBin()	Показать изображение двоичного значения объекта
ShowOct()	Показать изображение восьмеричного значения объекта
ShowDec()	Показать изображение десятичного значения объекта
ShowHex()	Показать изображение шестнадцатеричного значения объекта
operator +(T& s1. T& s2)	Для объектов SymbString — конкатенация строк s1 и s2; для объектов прочих классов — сложение соответствующих численных значений с последующим преобразованием к типу T
operator - (T& s1. T& s2)	Для объектов SymbString — если s2 содержится как подстрока в s1, то результатом является строка, полученная из s1 удалением подстроки s2; в противном случае возвращается значение s1; для объектов прочих классов — вычитание соответствующих численных значений с последующим преобразованием к типу T

**Примечание:** Первые четыре операции могут применяться к объектам любых классов, за исключением класса SymbString.

**Табл 1.3.** Спецификация вариантов

Вариант	T1	T2	Операции(методы)
1	SymbString	BinString	ShowOct(). ShowDec(). ShowHex()
2	SymbString	BinString	operator +(T& s1. T& s2)
3	SymbString	BinString	operator - (T& s1. T& s2)
4	SymbString	OctString	operator +(T& s1. T& s2)
5	SymbString	OctString	operator - (T& s1. T& s2)
6	SymbString	DecString	ShowBin(). ShowOct(). ShowHex()
7	SymbString	DecString	operator +(T& s1. T& s2)
8	SymbString	DecString	operator - (T& s1. T& s2)
9	SymbString	HexString	operator +(T& s1. T& s2)
10	SymbString	HexString	operator - (T& s1. T& s2)

**Табл. 1.1.** Перечень типов объектов

Класс	Объект
Triangle	Треугольник
Quadrate	Квадрат
Rectangle	Прямоугольник
Tetragon	Четырехугольник
Pentagon	Пятиугольник

**Табл. 1.2.** Перечень дополнительных операций (методов)

Операция (метод)	Описание
Move()	Переместить объект на плоскости
Compare(T& ob1, T& об2)	Сравнить объекты ob1 и об2 по площади
IsIntersect(T& об2, T& об2)	Определить факт пересечения объектов об1 и об2 (есть пересечение или нет)
IsInclucle(T& об1, T& об2)	Определить факт включения объекта об2 в объект об1

**Табл 1.3.** Спецификация вариантов

Вариант	T1	T2	Операции(методы)
11	Triangle	Quadrate	Move().Compare(T&, T&)
12	Quadrate	Pentagon	Move().IsIntersect(T&, T&)
13	Triangle	Rectangle	Move().Compare(T&, T&)
14	Triangle	Rectangle	Move().IsIntersect(T&, T&)
15	Rectangle	Pentagon	Move().IsInclucle(T& , T&)
16	Triangle	Tetragon	Move().Compare(T&, T&)
17	Triangle	Tetragon	Move().IsIntersect(T&, T&)
18	Triangle	Tetragon	Move().IsInclucle(T& , T&)
19	Triangle	Pentagon	Move().Compare(T&, T&)
20	Triangle	Pentagon	Move().IsInclucle(T& , T&)

## Вариант 1

Известный художник решил написать новый шедевр. После многих дней усердной работы он захотел исследовать свое творение. Художник вспомнил, что картина писалась следующим образом: сначала был взят белый холст, имеющий форму прямоугольника шириной  $w$  и высотой  $h$ . Затем художник нарисовал на этом холсте  $n$  прямоугольников со сторонами, параллельными сторонам холста и вершинами, расположенными в целочисленных координатах. Помогите художнику определить площадь не закрашенной части холста.

### Входные данные

Первая строка входного файла INPUT.TXT содержит два натуральных числа  $w$  и  $h$  ( $1 \leq w, h \leq 100$ ). Во второй строке записано целое число  $n$  ( $0 \leq n \leq 5000$ )

– количество прямоугольников. Следующие  $n$  строк содержат информацию о всех прямоугольниках. Каждая строка описывает один прямоугольник в виде четырех чисел  $x_1, y_1, x_2, y_2$ , где  $(x_1, y_1)$  и  $(x_2, y_2)$  – координаты левого верхнего и правого нижнего угла прямоугольника соответственно.

Выходные данные

Выведите в выходной файл OUTPUT.TXT одно целое число – площадь не закрашенной части холста.

## Вариант 2

Одним из интересных объектов, изучаемых в теории игр, являются так называемые антагонистические игры двух лиц. Такие игры характеризуются множеством  $X$  стратегий первого игрока, множеством  $Y$  стратегий второго игрока и функцией выигрыша  $K(x, y)$  ( $x$  из  $X$ ,  $y$  из  $Y$ ). Если множества стратегий  $X$  и  $Y$  конечны, то такую игру принято называть матричной, так как функцию выигрыша  $K$  в этом случае удобно задавать матрицей.

Рассмотрим матричную игру, в которой  $X = \{1, \dots, n\}$ ,  $Y = \{1, \dots, m\}$ . Матрицу выигрышей обозначим символом  $K$ . Нижним значением игры назовем число  $\min_{j=1..m} \max_{i=1..n} K_{ij}$ . Верхним значением игры назовем число  $\max_{j=1..m} \min_{i=1..n} K_{ij}$ . Отметим также, что игры, у которых нижнее и верхнее значение совпадают, называются играми с седловой точкой.

Задана матрица выигрышей  $K$  для некоторой матричной игры. Найдите ее верхнее и нижнее значение.

Входные данные

Первая строка входного файла INPUT.TXT содержит целые числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ). Далее следуют  $n$  строк по  $m$  чисел в каждой.  $j$ -ое число  $i$ -ой строки равно  $K_{ij}$ . Все  $K_{ij}$  по модулю не превосходят 1000.

Выходные данные

В выходной файл OUTPUT.TXT выведите нижнее и верхнее значение игры через пробел.

## Вариант 3

Рассмотрим таблицу, содержащую  $n$  строк и  $m$  столбцов, в каждой клетке которой расположен ноль или единица. Назовем такую таблицу симпатичной, если в ней нет ни одного квадрата  $2$  на  $2$ , заполненного целиком нулями или целиком единицами.

Так, например, таблица 4 на 4, расположенная слева, является симпатичной, а расположенная справа таблица 3 на 3 - не является.

1	0	1	0
1	1	1	0
0	1	0	1
0	0	0	0

0	0	1
0	0	1
1	1	1

Задано несколько таблиц. Необходимо для каждой из них выяснить, является ли она симпатичной.

Входные данные

Первая строка входного файла INPUT.TXT содержит количество  $t$  ( $1 \leq t \leq 10$ ) наборов входных данных. Далее следуют описания этих наборов. Описание каждого набора состоит из строки, содержащей числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ), и  $n$  строк, каждая из которых содержит по  $m$  чисел, разделенных пробелами.  $j$ -ое число в  $i+1$ -ой строке описания набора входных данных - элемент  $a_{ij}$  соответствующей таблицы. Гарантируется, что все  $a_{ij}$  равны либо нулю, либо единице.

Выходные данные

Для каждого набора входных данных выведите в файл OUTPUT.TXT единственную строку, содержащую слово «YES», если соответствующая таблица является симпатичной, и слово «NO» - в противном случае.

#### Вариант 4

Рассмотрим бесконечный лист клетчатой бумаги. Закрасим некоторое множество клеток в черный цвет. Теперь мы хотим закрасить минимальное количество клеток, так, чтобы множество черных клеток стало выпуклым.

Напомним, что геометрическая фигура  $\Phi$  называется выпуклой, если для любых точек  $A$  из  $\Phi$  и  $B$  из  $\Phi$  с вещественными координатами отрезок  $[AB]$  принадлежит  $\Phi$ .

Входные данные

В первой строке входного файла INPUT.TXT содержатся два числа  $N$  и  $M$  ( $1 \leq N, M \leq 100$ ) — размеры куска бумаги, куда попали все черные клетки. В каждой из следующих  $N$  строк содержится  $M$  символов «\*» или «.»». Символ «\*» обозначает черную клетку, а «.» белую.

Выходные данные

В выходной файл OUTPUT.TXT выведите выпуклое множество, содержащее минимальное количество дополнительно покрашенных черных клеток, в ровно N строках по M символов «\*» или «.» в каждой.

### Вариант 5

Судoku размера n называется квадрат со стороной  $n^2$ , разделенный на  $n^2$  средних квадратов со стороной n, каждый из которых разделен на  $n^2$  маленьких квадратов. В каждом маленьком квадрате записано число от 1 до  $n^2$ .

Судoku называется правильным, если в каждом столбце, каждой строке и каждом среднем квадрате встречаются все числа от 1 до  $n^2$ .

Недавно Вася нарисовал Судoku размера n. Ваша задача – помочь ему определить правильный ли он.

Входные данные

В первой строке входного файла INPUT.TXT содержится число n ( $1 \leq n \leq 10$ ). В следующих  $n^2$  строчках содержится по  $n^2$  чисел, задающих нарисованный Васей Судoku.

Все числа во входном файле натуральные и не превосходят 100 по модулю.

Выходные данные

Если Судoku правильный, то выведите в выходной файл OUTPUT.TXT слово «Correct», иначе выведите «Incorrect».

### Вариант 6

Требуется вывести квадрат, состоящий из  $N \times N$  клеток, заполненных числами от 1 до  $N^2$  по спирали (см. примеры).

1	2	3	4	5
16	17	18	19	6
15	24	25	20	7
14	23	22	21	8
13	12	11	10	9

Входные данные

Во входном файле INPUT.TXT задано целое число N – размер квадратной матрицы ( $2 \leq N \leq 100$ ).

Выходные данные

В выходной файл OUTPUT.TXT выведите матрицу, заполненную числами от 1 до  $N^2$  по спирали, при этом между числами может быть любое количество пробелов. Не допускается начинать спираль в ином, кроме верхнего левого, углу, закручивать спираль против часовой стрелки или изнутри наружу.

## Вариант 7

Требуется заполнить змейкой квадратную матрицу так, как показано на рисунке справа: заполнение происходит с единицы из левого верхнего угла и заканчивается в правом нижнем числом  $N^2$ , где  $N$  – порядок матрицы.

1	3	4	10
2	5	9	11
6	8	12	15
7	13	14	16

Входные данные

Во входном файле INPUT.TXT задано натуральное число  $N$  – размер квадратной матрицы ( $N \leq 100$ ).

Выходные данные

В выходной файл OUTPUT.TXT выведите матрицу, заполненную числами от 1 до  $N^2$  змейкой, при этом между числами может быть любое количество пробелов.

## Вариант 8

Требуется решить невырожденную систему, состоящую из  $N$  линейных уравнений с  $N$  неизвестными:

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \\ a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n = b_i \\ \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases}$$

Входные данные

В первой строке входного файла INPUT.TXT задано натуральное число  $N$  – ранг системы, далее следуют  $N$  строк, каждая из которых состоит из  $N+1$  целых чисел: коэффициенты  $i$ -й строки уравнения –  $N$  чисел  $a_{ij}$  и  $b_i$ . ( $N \leq 100$ ,  $|a_{ij}| < 10$ ,  $|b_i| < 104$ ). Числа разделены одним или несколькими пробелами. Перед первым числом строки может быть ноль, один или несколько пробелов.

Выходные данные

В единственную строку выходного файла OUTPUT.TXT нужно вывести через пробел корни приведенной во входном файле системы линейных уравнений. Корни следует выводить в порядке возрастания их номеров ( $x_1, x_2, \dots, x_n$ ). Гарантируется, что все корни целые и не превосходят значения 10 по абсолютной величине.

## Вариант 9

Мальчику Васе очень нравится известная игра "Сапер" ("Minesweeper").

В "Сапер" играет один человек. Игра идет на клетчатом поле (далее будем называть его картой)  $N \times M$  ( $N$  строк,  $M$  столбцов). В  $K$  клетках поля стоят мины, в остальных клетках записано либо число от 1 до 8 — количество мин в соседних клетках, либо ничего не написано, если в соседних клетках мин нет. Клетки являются соседними, если они имеют хотя бы одну общую точку, в одной клетке не может стоять более одной мины. Изначально все клетки поля закрыты. Игрок за один ход может открыть какую-нибудь клетку. Если в открытой им клетке оказывается мина — он проигрывает, иначе игроку показывается число, которое стоит в этой клетке, и игра продолжается. Цель игры — открыть все клетки, в которых нет мин.

У Васи на компьютере есть эта игра, но ему кажется, что все карты, которые в ней есть, некрасивые и неинтересные. Поэтому он решил нарисовать свои. Однако фантазия у него богатая, а времени мало, и он хочет успеть нарисовать как можно больше карт. Поэтому он просто выбирает  $N$ ,  $M$  и  $K$  и расставляет мины на поле, после чего все остальные клетки могут быть однозначно определены. Однако на определение остальных клеток он не хочет тратить свое драгоценное время. Помогите ему!

По заданным  $N$ ,  $M$ ,  $K$  и координатам мин восстановите полную карту.

Входные данные

В первой строке входного файла INPUT.TXT содержатся числа  $N$ ,  $M$  и  $K$  ( $1 \leq N \leq 200$ ,  $1 \leq M \leq 200$ ,  $0 \leq K \leq N \times M$ ). Далее идут  $K$  строк, в каждой из которых содержится по два числа, задающих координаты мин. Первое число в каждой строке задает номер строки клетки, где находится мина, второе число — номер столбца. Левая верхняя клетка поля имеет координаты  $(1,1)$ , правая нижняя — координаты  $(N,M)$ .

Выходные данные

Выходной файл OUTPUT.TXT должен содержать  $N$  строк по  $M$  символов — соответствующие строки карты.  $j$ -й символ  $i$ -й строки должен содержать символ '\*' (звездочка) если в клетке  $(i,j)$  стоит мина, цифру от 1 до 8, если в этой клетке стоит соответствующее число, либо '.' (точка), если клетка  $(i,j)$  пустая.

**Вариант 10**

Из шахматной доски по границам клеток выпилили связную (не распадающуюся на части) фигуру без дыр. Требуется определить ее периметр.



Входные данные

Во входном файле INPUT.TXT сначала записано число  $N$  ( $1 \leq N \leq 64$ ) – количество выпиленных клеток. В следующих  $N$  строках указаны координаты выпиленных клеток, разделенные пробелом (номер строки и столбца – числа от 1 до 8). Каждая выпиленная клетка указывается один раз.

Выходные данные

В выходной файл OUTPUT.TXT выведите одно число – периметр выпиленной фигуры (сторона клетки равна единице).

### Вариант 11

Спецслужбы получили информацию о том, что в труднодоступной части Муравийской пустыни расположена хорошо замаскированная база террористов. В руки спецслужб попал и план этой базы, которая с большой высоты выглядит как группа скальных обломков, весьма часто встречающихся среди Муравийских песков. Президент отдал приказ уничтожить базу крылатыми ракетами. Ваша задача - по карте пустыни, полученной со спутника и плану базы определить количество возможных положений базы террористов. Помните, что террористы могли привезти на территорию базы камни!

Входные данные

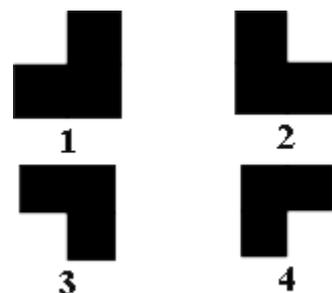
В первой строке файла INPUT.TXT записаны числа  $N_b$  и  $M_b$  ( $1 \leq N_b, M_b \leq 20$ ). В следующих  $N_b$  строках записан план базы. Каждая из этих строк содержит по  $M_b$  символов "#" (ASCII 35) или "." (ASCII 46). Символ "#" обозначает фрагмент базы, а символ "." - песок. В следующей строке записаны числа  $N_d$  и  $M_d$  ( $1 \leq N_d, M_d \leq 100$ ). И остаток файла содержит карту участка пустыни на котором, предположительно, находится база террористов -  $N_d$  строк по  $M_d$  символов "#" или "." в каждой.

Выходные данные

Запишите в файл OUTPUT.TXT количество возможных положений базы террористов.

### Вариант 12

Вы являетесь одним из разработчиков нового архитектурного пакета прикладных программ «CadArch». Одной из его функций является проектирование укладки половых плиток. В настоящее время вы занимаетесь программной реализацией модуля, который отвечает за укладку плиток в прямоугольных помещениях.



Для простоты будем считать, что пол помещения представляет собой прямоугольник размером  $n$  на  $m$  метров, разбитый на  $m \cdot n$  квадратиков со стороной по 1 метру. Кроме этого, будем считать, что имеется четыре типа плиток, показанные в таблице. Каждая из плиток представляет собой квадрат размером 2 на 2 метра, из которого вырезан один квадратик размером 1 на 1 метр.

Проектируемый модуль должен работать следующим образом. На вход модуля подается набор команд, каждая из которых обозначает, в какое место и какого типа плитку необходимо положить. Команда обрабатывается следующим образом: если ни один из квадратиков, который должна занимать текущая плитка, не занят и плитка полностью помещается внутри прямоугольника, то плитка размещается в указанном месте, в противном случае – нет.

Требуется написать программу, которая определяет, какая площадь в соответствии с заданным набором команд будет покрыта плитками.

#### Входные данные

Первая строка входного файла INPUT.TXT содержит два числа  $n$  и  $m$  — длина и ширина пола помещения ( $1 \leq m, n \leq 50$ ). Вторая строка содержит число  $k$  — количество команд, которые необходимо обработать. Каждая из последующих  $k$  строк содержит описание одной команды из набора команд. Описание команды состоит из трех чисел. Первое число определяет тип плитки (число от 1 до 4), а два других - координаты левого верхнего угла квадрата  $(y, x)$  размером 2 на 2, в который вписана соответствующая плитка ( $0 \leq x, y, k \leq 1000$ ).

#### Выходные данные

В выходной файл OUTPUT.TXT необходимо вывести одно число, определяющее площадь, покрытую плитками после выполнения заданной во входном файле последовательности команд.

### Вариант 13

В данной двумерной целочисленной таблице размером  $N \times N$  требуется найти три элемента, сумма которых максимальна. При этом первый элемент должен

быть соседним по горизонтали или вертикали со вторым, а второй — с третьим.

Входные данные

Входной файл INPUT.TXT содержит в первой строке число  $N$  ( $1 < N \leq 2000$ ). В следующих  $N$  строках записано по  $N$  чисел – элементы таблицы. Элементы матрицы по абсолютной величине не превышают 100.

Выходные данные

Выходной файл OUTPUT.TXT должен содержать единственное число — максимальную сумму.

### **Вариант 14**

Задана матрица, содержащая  $N$  строк и  $M$  столбцов. Седловой точкой этой матрицы назовем элемент, который одновременно является минимумом в своей строке и максимумом в своем столбце.

Найдите количество седловых точек заданной матрицы.

Входные данные

Входной файл INPUT.TXT в первой строке содержит целые числа  $N$  и  $M$  ( $1 \leq N, M \leq 750$ ). Далее следуют  $N$  строк по  $M$  чисел в каждой. Элементы матрицы - целые числа, не превосходящие 1000 по абсолютной величине. Числа могут быть разделены несколькими пробелами.

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

### **Вариант 15**

Со спутника-шпиона получено изображение в некотором волновом диапазоне сверхсекретной военной базы предполагаемого противника. База расположена на Антарктиде, все постройки на ней высечены из кубов льда и имеют на фотографии квадратную форму и не имеют общих фрагментов стен ненулевой длины (по всей видимости, это сделано в целях маскировки от локаторов, работающих в инфракрасном спектре). Благодаря мастерству операторов оказалось, что стены разных построек параллельны границам фотографии.

Для того, чтобы составить сверхсрочный отчет для командования, необходимо узнать, сколько зданий находятся на базе. Напишите программу, которая это сделает.

Входные данные

В первой строке входного файла INPUT.TXT записаны числа  $N$  и  $M$  ( $1 \leq M, N \leq 500$ ) – размеры фотографии в пикселях по вертикали и по горизонтали. Следующие  $N$  строк содержат по  $M$  символов каждая: символ '.' соответствует пустому месту, '#' – элементу постройки.

Выходные данные

В выходной файл OUTPUT.TXT выведите единственное число – количество построек на базе.

### Вариант 16

На хоккейном стадионе в одном большом городе расположено большое прямоугольное табло. Оно имеет  $n$  строк и  $m$  столбцов (то есть состоит из  $n \times m$  ячеек). Во время хоккейного матча это табло служит для отображения счета и времени, прошедшего с начала тайма, а в перерывах на нем показывают различную рекламу.

В связи с этим возникла задача проверки возможности показа на этом табло определенной рекламной заставки. Заставка также, как и табло, имеет размер  $n$  строк на  $m$  столбцов. Каждая из ячеек заставки окрашена в один из четырех цветов - трех основных: красный - R, зеленый - G, синий - B и черный - .(точка).

Каждая из ячеек табло характеризуется своими цветопередаточными возможностями. Любая из ячеек табло может отображать черный цвет - это соответствует тому, что на нее вообще не подается напряжение. Также каждая из ячеек может отображать некоторое подмножество множества основных цветов. В этой задаче эти подмножества будут кодироваться следующим образом:

- 0 - ячейка может отображать только черный цвет;
- 1 - ячейка может отображать только черный и синий цвета;
- 2 - ячейка может отображать только черный и зеленый цвета;
- 3 - ячейка может отображать только черный, зеленый и синий цвета;
- 4 - ячейка может отображать только черный и красный цвета;
- 5 - ячейка может отображать только черный, красный и синий цвета;
- 6 - ячейка может отображать только черный, красный и зеленый цвета;
- 7 - ячейка может отображать только черный, красный, зеленый и синий цвета.

Напишите программу, которая по описанию табло и заставки определяет: возможно ли на табло отобразить эту заставку.

Входные данные

Первая строка входного файла INPUT.TXT содержит целые числа  $n$  и  $m$  ( $1 \leq n$ ,  $m \leq 100$ ). Далее идут  $n$  строк по  $m$  символов каждая - описание заставки. Каждый из символов описания заставки принадлежит множеству  $\{R, G, B, .\}$ . Их значения описаны выше.

После этого идет описание табло. Оно содержит  $n$  строк по  $m$  чисел, разделенных пробелами. Значения чисел описаны выше.

Выходные данные

В выходной файл OUTPUT.TXT выведите YES, если на табло возможно отобразить заставку и NO - в противном случае.

### Вариант 17

«Морской бой» - игра для двух участников, в которой игроки по очереди называют координаты на неизвестной им карте соперника. Если у соперника по этим координатам имеется корабль, то корабль или его часть «топится», а попавший получает право сделать еще один ход. Цель игрока - первым поразить все корабли противника.

«Морской бой» очень популярен среди учеников одной физико-математической школы. Ребята очень любят в него играть на переменах. Вот и сейчас ученики Иннокентий и Емельян начали новую партию.

Правила, по которым ребята расставляют корабли перед началом партии, несколько отличаются от классических. Во-первых, игра происходит на поле размером  $N \times M$ , а не  $10 \times 10$ . Во-вторых, число кораблей, их размер и форма выбираются ребятами перед партией - так играть намного интереснее.

Емельян уже расставил все свои корабли, кроме одного однопалубного. Такой корабль занимает ровно одну клетку.

Задана расстановка кораблей Емельяна. Найдите число способов поставить оставшийся однопалубный корабль. При этом учитывайте, что по правилам его можно ставить только в ту клетку, все соседние с которой не заняты. В этой задаче соседними считаются клетки, имеющие общую сторону.

Входные данные

Первая строка входного файла INPUT.TXT содержит два числа:  $N$  и  $M$  ( $1 \leq N$ ,  $M \leq 100$ ). Последующие  $N$  строк описывают игровое поле - каждая из них содержит  $M$  символов. Символом «.» (точка) обозначена свободная клетка, символом «\*» (звездочка) - занятая кораблем.

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

## Вариант 18

Администрация города подбирает площадку для строительства новых спортивных сооружений. На рассмотрении несколько проектов, каждый проект требует выделения некоторого прямоугольного участка земли. Некоторые участки оказались частично или полностью в пользовании частными лицами, а в случае утверждения проекта администрация будет вынуждена выкупить этот участок, поэтому для определения стоимости очень важно знать площадь пересечения участков. Участки частников также прямоугольной формы (рис. 1) и стороны всех участков параллельны координатным осям. Для каждого проекта был построен план, включающий подобранный участок и его окружение. В приведенном примере показано пересечение участков частников (тонкая линия) с участком, подобранным для строительства (толстая линия). Помогите определить суммарную площадь пересечения участков частников с участком, выбранным для строительства. При этом следует учесть, что выкупаемая земля может принадлежать сразу нескольким участникам и в этом случае необходимо выкупать землю многократно (одна и та же площадь земли может быть посчитана несколько раз).

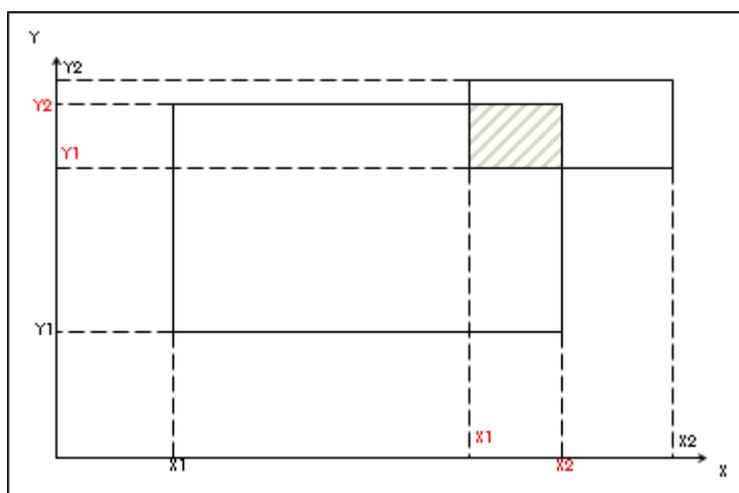


Рис. 1.

### Входные данные

В первой строке входного файла INPUT.TXT содержится число  $N$  - количество участков частников, отображенных на плане. Затем следуют  $N$  строк с координатами двух вершин этих прямоугольных участков. В последней строке плана координаты участка, выбранного для строительства. Координаты противоположных вершин одного прямоугольника описываются в формате  $X1 Y1 X2 Y2$ . Координатами вершин являются целые, неотрицательные числа, не больше 100. Количество исходных прямоугольников не больше 20.

### Выходные данные

В единственную строку выходного файла OUTPUT.TXT следует вывести одно целое число – искомую суммарную площадь пересечения прямоугольников.

### Вариант 19

Одна из базовых задач компьютерной графики – обработка черно-белых изображений. Изображения можно представить в виде прямоугольников шириной  $w$  и высотой  $h$ , разбитых на  $w \times h$  единичных квадратов, каждый из которых имеет либо белый, либо черный цвет. Такие единичные квадраты называются пикселями. В памяти компьютера сами изображения хранятся в виде прямоугольных таблиц, содержащих нули и единицы.

Во многих областях очень часто возникает задача комбинации изображений. Одним из простейших методов комбинации, который используется при работе с черно-белыми изображениями, является попиксельное применение некоторой логической операции. Это означает, что значение пиксела результата получается применением этой логической операции к соответствующим пикселям аргументов. Логическая операция от двух аргументов обычно задается таблицей истинности, которая содержит значения операции для всех возможных комбинаций аргументов.

Например, для операции «исключающее ИЛИ» эта таблица выглядит так:

Первый аргумент	Второй аргумент	Результат
0	0	0
0	1	1
1	0	1
1	1	0

Требуется написать программу, которая вычислит результат попиксельного применения заданной логической операции к двум черно-белым изображениям одинакового размера.

#### Входные данные

Первая строка входного файла INPUT.TXT содержит два целых числа  $w$  и  $h$  ( $1 \leq w, h \leq 100$ ). Последующие  $h$  строк описывают первое изображение и каждая из этих строк содержит  $w$  символов, каждый из которых равен нулю или единице. Далее следует описание второго изображения в аналогичном формате. Последняя строка входного файла содержит описание логической операции в виде четырех чисел, каждое из которых – ноль или единица. Первое из них есть результат применения логической операции в случае, если

оба аргумента – нули, второе – результат в случае, если первый аргумент – ноль, второй – единица, третье – результат в случае, если первый аргумент – единица, второй – ноль, а четвертый – в случае, если оба аргумента – единицы.

**Выходные данные**

В выходной файл OUTPUT.TXT необходимо вывести результат применения заданной логической операции к изображениям в том же формате, в котором изображения заданы во входном файле.

## **Вариант 20**

Аня недавно узнала, что такое квадратная матрица размерности  $n$ . Это таблица  $n \times n$  с целыми числами в ячейках. Число, стоящее на пересечении  $i$ -ой строки и  $j$ -ого столбца матрицы  $A$ , кратко обозначается  $A[i, j]$ . Матрицы можно умножать, и Аня быстро освоила, как запрограммировать эту операцию с помощью циклов. Результатом умножения двух матриц  $A$  и  $B$  будет матрица  $C$ , элементы которой определяются следующим образом:

$$C[i, j] = \sum_{k=1}^n A[i, k]B[k, j]$$

Матрицы ей понадобились для конкретной задачи, в которой надо узнать определенный элемент произведения нескольких матриц. Это уже достаточно сложная задача для Ани, но она усложняется тем, что все вычисления ведутся по модулю некоторого простого числа  $p$ , то есть если при арифметических операциях получается число, большее, либо равное  $p$ , оно заменяется на остаток при делении на  $p$ .

Помогите Ане вычислить нужный ей элемент.

**Входные данные**

В первой строчке входного файла INPUT.TXT стоят два числа:  $m$  - количество матриц,  $n$  - размер каждой из матриц ( $1 \leq m \leq 130$ ,  $1 \leq n \leq 130$ ). В следующей строчке содержатся номер строки и столбца, интересующего Аню элемента  $1 \leq a \leq n$ ,  $1 \leq b \leq n$ . В третьей строке содержится простое число  $p \leq 1000$ . Далее следует описание  $m$  матриц. Описание каждой матрицы состоит из  $n$  строк. В каждой из строк содержится  $n$  неотрицательных целых чисел, меньших  $p$ . Соседние числа в строке разделены пробелом, а перед каждой матрицей пропущена строка.

**Выходные данные**

В выходной файл OUTPUT.TXT выведите нужный Ане элемент произведения матриц.

## **Вариант 21**

На сайте сотового оператора VeerLine сделали защиту от роботов, рассылающих SMS-сообщения: прежде, чем отправить SMS, пользователь должен написать, какую фигуру он видит в специальном окошке: квадрат или круг. Причем, для усиления защиты, в рисунок внесены небольшие помехи.

Коле срочно нужно разослать всем друзьям сообщение, поэтому он просит Вас написать программу, распознающую изображение.

Экспериментально установлено, что система рисует квадрат с помехами следующим образом: сначала на белом фоне рисуется черный квадрат  $k \times k$  клеток ( $k \geq 3$ ), затем некоторые клетки на границе квадрата (на рисунке обозначены цифрой 1) закрашиваются белым, а некоторые клетки (если таковые существуют), граничащие с квадратом (на рисунке обозначены цифрой 2), закрашиваются черным.

	2	2	2	2	2	2	
	2	1	1	1	1	2	
	2	1			1	2	
	2	1			1	2	
	2	1	1	1	1	2	
	2	2	2	2	2	2	

Например, квадрат  $4 \times 4$  после нанесения помех может выглядеть так:

	*			*	*		
		*	*			*	
	*	*	*	*	*	*	
		*			*		
	*				*		

Входные данные

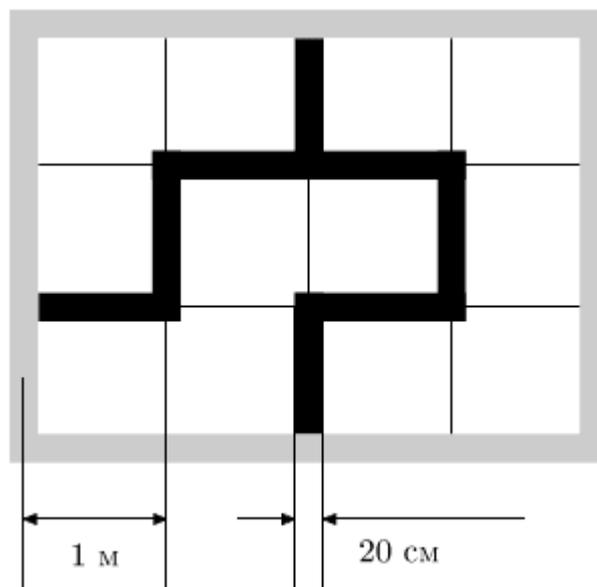
Первая строка входного файла INPUT.TXT содержит целые числа  $n$  и  $m$  - размеры экрана ( $1 \leq n, m \leq 1000$ ). Следующие  $n$  строк, по  $m$  символов в каждой, содержат описание картинки. Черные клетки обозначены символом «\*», а белые - символом «.».

## Выходные данные

В выходной файл OUTPUT.TXT выведите «SQUARE», если заданная картинка может быть квадратом после преобразований, описанных в условии, иначе выведите слово «CIRCLE».

### Вариант 22

Фирма, в которой вы работаете, исполняет проект строительства суперсовременного бизнес-центра. И вот, когда заказ на склад строительных материалов почти готов к подписанию директором, оказывается, что туда забыли включить некоторые товары. Конкретно, в нем не были предусмотрены материалы, необходимые для возведения стен между отдельными секциями в подвальных помещениях. Необходимо срочно написать программу, которая сможет рассчитать необходимое количество материалов.



На плане подвальный этаж имеет вид прямоугольника, стороны которого лежат на линиях сетки с квадратными клетками. Сетка имеет такой масштаб, что сторона клетки соответствует одному метру стены подвального этажа. Каждая клетка на плане целиком принадлежит одному из помещений. Для каждой пары соседних по стороне клеток, отнесенных к разным помещениям, вдоль всей их общей стороны должна быть возведена стена толщиной 20 сантиметров и высотой три метра. Материалы для постройки всех внешних стен подвального этажа уже включены в заказ.

## Входные данные

Первая строка входного файла INPUT.TXT содержит два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 100$ ), разделенных пробелами - размеры подвала на плане. Каждая из  $N$  последующих строк содержит по  $M$  натуральных чисел, не превосходящих  $M \times N$ , задающих номер помещения, к которому относится данная клетка. Эти числа разделены пробелами.

## Выходные данные

В выходной файл OUTPUT.TXT выведите одно вещественное число с точностью не менее 0.001 - общий объем возводимых стен в кубических метрах.

## Вариант 23

Миша уже научился хорошо фотографировать и недавно увлекся программированием. Первая программа, которую он написал, позволяет формировать негатив бинарного черно-белого изображения.

Бинарное черно-белое изображение – это прямоугольник, состоящий из пикселей, каждый из которых может быть либо черным, либо белым. Негатив такого изображения получается путем замены каждого черного пикселя на белый, а каждого белого пикселя – на черный.

Миша, как начинающий программист, написал свою программу с ошибкой, поэтому в результате ее исполнения мог получаться некорректный негатив. Для того чтобы оценить уровень несоответствия получаемого негатива исходному изображению, Миша начал тестировать свою программу.

В качестве входных данных он использовал исходные изображения. Сформированные программой негативы он начал тщательно анализировать, каждый раз определяя число пикселей негатива, которые получены с ошибкой.

Требуется написать программу, которая в качестве входных данных использует исходное бинарное черно-белое изображение и полученный Мишиной программой негатив, и на основе этого определяет количество пикселей, в которых допущена ошибка.

### Входные данные

Первая строка входного файла INPUT.TXT содержит целые числа  $n$  и  $m$  ( $1 \leq n, m \leq 100$ ) – высоту и ширину исходного изображения (в пикселях). Последующие  $n$  строк содержат описание исходного изображения. Каждая строка состоит из  $m$  символов «В» и «W». Символ «В» соответствует черному пикселю, а символ «W» – белому. Далее следует пустая строка, а после нее – описание выведенного Мишиной программой изображения в том же формате, что и исходное изображение.

### Выходные данные

В выходной файл OUTPUT.TXT необходимо вывести число пикселей негатива, которые неправильно сформированы Мишиной программой.

## Вариант 24

Магический квадрат - это квадратная таблица  $N \times N$ , заполненная  $N^2$  числами от 1 до  $N^2$  таким образом, что сумма чисел в каждой строке, каждом столбце и на обеих диагоналях одинакова. При этом числа в таблице не должны повторяться и каждое из чисел от 1 до  $N^2$  должно в ней присутствовать.

4	9	2
3	5	7
8	1	6

Требуется написать программу, которая по заданному  $N$  строит магический квадрат.

Входные данные

Во входном файле INPUT.TXT задано единственное натуральное число  $N$  - размерность магического квадрата ( $N \leq 1000$ ).

Выходные данные

В выходной файл OUTPUT.TXT необходимо вывести магический квадрат - матрицу, состоящую из  $N$  строк по  $N$  элементов в каждой. Если вариантов решения несколько, то можно вывести любой. В том случае, когда решение не существует, следует вывести "Impossible".

## Вариант 25

Матрица  $A$  считается меньше (больше) матрицы  $B$ , если при просмотре слева направо и сверху вниз после всех равенств элементов матриц следующий элемент из матрицы  $A$  меньше (больше), чем соответствующий элемент из  $B$ . Такое сравнение называется лексикографическим и напоминает способ сравнения слов в словаре.

Дана прямоугольная матрица  $A$  размером  $n \times m$ , все элементы которой различны. В ней можно менять местами два произвольных столбца, а также менять местами две произвольных строки.

Пусть  $A_{max}$  - максимальная матрица, получаемая из исходной матрицы путем любого требуемого количества вышеприведенных операций. Необходимо написать программу, находящую последний (правый нижний) элемент из  $A_{max}$ .

Входные данные

В первой строке входного файла INPUT.TXT записаны целые числа  $n, m$  ( $1 \leq n, m \leq 400$ ). Далее в  $n$  строках содержится сама матрица. Каждая из этих строк содержит  $m$  целых чисел от 0 до 2147483647. Все элементы матрицы различны.

Выходные данные

В выходной файл OUTPUT.TXT выведите искомый правый нижний элемент матрицы  $A_{max}$ .

## Вариант 26

Игра «Жизнь» была придумана английским математиком Джоном Конвейем в 1970 году. Впервые описание этой игры опубликовано в октябрьском выпуске (1970) журнала *Scientific American*, в рубрике «Математические игры» Мартина Гарднера.

Место действия этой игры – «вселенная» – это размеченная на клетки поверхность. Каждая клетка на этой поверхности может находиться в двух состояниях: быть живой или быть мертвой. Клетка имеет восемь соседей. Распределение живых клеток в начале игры называется первым поколением. Каждое следующее поколение рассчитывается на основе предыдущего по таким правилам:

пустая (мертвая) клетка с ровно тремя живыми клетками-соседями оживает;

если у живой клетки есть две или три живые соседки, то эта клетка продолжает жить; в противном случае (если соседок меньше двух или больше трех) клетка умирает (от «одиночества» или от «перенаселенности»).

В этой задаче рассматривается игра «Жизнь» на торе. Представим себе прямоугольник размером  $n$  строк на  $m$  столбцов. Для того, чтобы превратить его в тор мысленно «склеим» его верхнюю сторону с нижней, а левую с правой.

Таким образом, у каждой клетки, даже если она раньше находилась на границе прямоугольника, теперь есть ровно восемь соседей.

Ваша задача состоит в том, чтобы найти конфигурацию клеток, которая будет через  $k$  поколений от заданного.

Входные данные

Первая строка входного файла `INPUT.TXT` содержит три целых числа:  $n$ ,  $m$ ,  $k$  ( $4 \leq n$ ,  $m \leq 100$ ;  $0 \leq k \leq 100$ ). Последующие  $n$  строк содержат по  $m$  символов каждая и описывают начальную конфигурацию.  $j$ -ый символ  $i$ -ой строки равен «.» (точка), если соответствующая клетка мертва, и «\*» (звездочка) – если жива.

Выходные данные

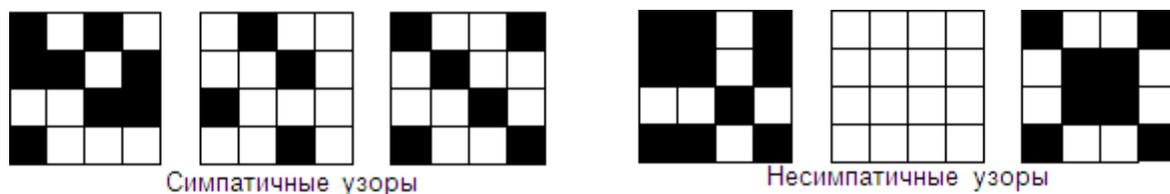
В выходной файл `OUTPUT.TXT` выведите конфигурацию клеток через  $k$  поколений после начального в том же формате, в каком конфигурация задается во входном файле.

## Вариант 27

На днях Иван у себя в прихожей выложил кафель, состоящий из квадратных черных и белых плиток. Прихожая Ивана имеет квадратную форму  $4 \times 4$ ,

вмещающую 16 плиток. Теперь Иван переживает, что узор из плиток, который у него получился, может быть не симпатичным. С точки зрения дизайнера симпатичным узором считается тот, который не содержит в себе квадрата 2x2, состоящего из плиток одного цвета.

Примеры возможных узоров:



По заданному расположению плиток в прихожей Ивана требуется определить: является ли выполненный узор симпатичным.

Входные данные

Входной файл INPUT.TXT содержит 4 строки по 4 символа «W» или «B» в каждой, описывающие узор из плиток. Символ «W» обозначает плитку белого цвета, а «B» - черного.

Выходные данные

В выходной файл OUTPUT.TXT выведите «Yes», если узор является симпатичным и «No» в противном случае.

## Вариант 28

Юный программист решил придумать собственную игру. Игра происходит на поле размером  $N \times N$  клеток, в некоторых клетках которого расположены города (каждый город занимает одну клетку; в каждой клетке может располагаться не более одного города). Всего должно быть чётное количество городов.

Изначально про каждую клетку игрового поля известно, расположен ли в ней город или нет. Чтобы начать игру, необходимо разделить игровое поле на два государства так, чтобы в каждом государстве было поровну клеток-городов.

Граница между государствами должна проходить по границам клеток таким образом, чтобы из любой клетки каждого государства существовал путь по клеткам этого же государства в любую другую его клетку (из клетки можно перейти в соседнюю, если они имеют общую сторону). Каждая клетка игрового поля должна принадлежать только одному из двух государств, при этом государства не обязаны состоять из одинакового количества клеток.

Требуется написать программу, которая с учетом сказанного разделит клетки заданного игрового поля между двумя государствами.

## Входные данные

Первая строка входного файла INPUT.TXT содержит одно натуральное число  $N$ , задающее размер игрового поля ( $N \leq 50$ ).

Последующие  $N$  строк содержат по  $N$  заглавных английских букв (без пробелов), кодирующих соответствующие клетки игрового поля: 'C' обозначает клетку, занятую городом, 'D' – пустую клетку. Гарантируется, что на поле есть хотя бы два города и всего их четное число.

## Выходные данные

Выходной файл OUTPUT.TXT должен содержать  $N$  строк по  $N$  цифр (без пробелов) в каждой, кодирующих соответствующие клетки. Цифра 1 обозначает, что данная клетка принадлежит первому государству, цифра 2 – данная клетка принадлежит второму государству. Если решений несколько, необходимо вывести любое из них.

## Вариант 29

Дано поле  $N \times M$  и объекты на нем в моменты времени  $T$  и  $T+1$ . Каждый объект представлен одной английской буквой и в любой момент времени может занимать ровно одну клетку поля.

Объект движется, если в два последовательных момента времени его положения различаются.

Найдите все, что движется.

## Входные данные

В первой строке входного файла INPUT.TXT записаны два целых числа  $N$  и  $M$  ( $1 \leq N, M \leq 100$ ). В следующих  $N$  строках по  $M$  символов – поле в момент времени  $T$ . Каждый символ либо является точкой «.», и это означает, что в этом месте поля ничего нет, либо английская буква, обозначающая то, что в этом месте находится объект. Никакие два различных объекта не обозначены одним и тем же символом.

Далее идет пустая строка.

В следующих  $N$  строках находится описание того же поля в момент времени  $T+1$  в том же формате. Множество объектов, находящихся на поле в момент времени  $T$ , равно множеству объектов в момент времени  $T + 1$ .

## Выходные данные

В первой строке выходного файла OUTPUT.TXT выведите количество движущихся объектов. Во второй строке выведите символы, соответствующие движущимся объектам, в алфавитном порядке, причем сначала выведите все

маленькие английские буквы, затем все большие. Пробелы между символами выводить не следует.

### **Вариант 30**

Секретарша Ирочка сегодня опоздала на работу и ей срочно нужно успеть к обеду сделать  $N$  копий одного документа. В ее распоряжении имеются два ксерокса, один из которых копирует лист за  $x$  секунд, а другой – за  $y$  секунд. (Разрешается использовать как один ксерокс, так и оба одновременно. Можно копировать не только с оригинала, но и с копии.) Помогите ей выяснить, какое минимальное время для этого потребуется.

Входные данные

Во входном файле INPUT.TXT записаны три натуральных числа  $N$ ,  $x$  и  $y$ , разделенные пробелом ( $1 \leq N \leq 2 \cdot 10^8$ ,  $1 \leq x, y \leq 10$ ).

Выходные данные

В выходной файл OUTPUT.TXT выведите одно число – минимальное время в секундах, необходимое для получения  $N$  копий.

### **Вариант 31**

В романе  $N$  глав. В  $i$ -той главе  $a_i$  страниц. Требуется издать роман в  $K$  томах так, чтобы объем самого «толстого» тома был минимален. В каждом томе главы располагаются по порядку своих номеров.

Требуется написать программу, которая найдет количество страниц в самом «толстом» томе.

Входные данные

Входной текстовый файл INPUT.TXT содержит в первой строке число  $N$  ( $1 \leq N \leq 100$ ). Во второй строке через пробел записаны  $N$  чисел – количество страниц в каждой главе. Количество страниц в романе не превышает 32767. В третьей строке записано число  $K$  ( $1 \leq K \leq N$ ).

Выходные данные

Выходной файл OUTPUT.TXT должен содержать количество страниц в самом «толстом» томе.

### **Вариант 32**

Когда Петя учился в школе, он часто участвовал в олимпиадах по информатике, математике и физике. Так как он был достаточно способным мальчиком и усердно учился, то на многих из этих олимпиад он получал дипломы. К окончанию школы у него накопилось  $n$  дипломов, причем, как оказалось, все они имели одинаковые размеры:  $w$  – в ширину и  $h$  – в высоту.

Сейчас Петя учится в одном из лучших российских университетов и живет в общежитии со своими одногруппниками. Он решил украсить свою комнату, повесив на одну из стен свои дипломы за школьные олимпиады. Так как к бетонной стене прикрепить дипломы достаточно трудно, то он решил купить специальную доску из пробкового дерева, чтобы прикрепить ее к стене, а к ней – дипломы. Для того чтобы эта конструкция выглядела более красиво, Петя хочет, чтобы доска была квадратной и занимала как можно меньше места на стене. Каждый диплом должен быть размещен строго в прямоугольнике размером  $w$  на  $h$ . Прямоугольники, соответствующие различным дипломам, не должны иметь общих внутренних точек.

Требуется написать программу, которая вычислит минимальный размер стороны доски, которая потребуется Пете для размещения всех своих дипломов.

Входные данные

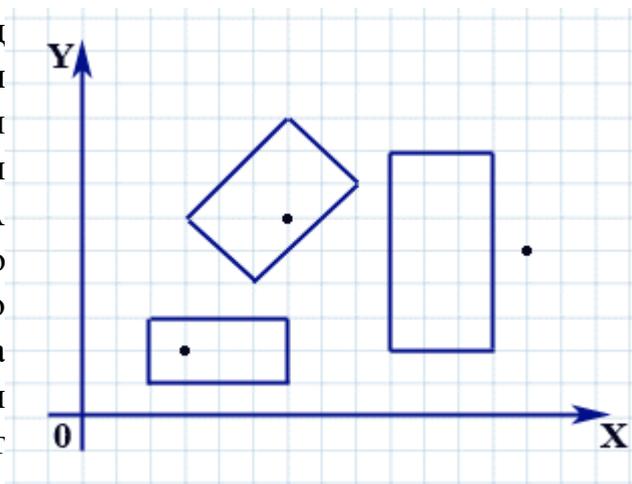
Входной файл INPUT.TXT содержит три целых числа:  $w$ ,  $h$ ,  $n$  ( $1 \leq w, h, n \leq 109$ ).

Выходные данные

В выходной файл OUTPUT.TXT выведите ответ на задачу.

### Вариант 33

Всем известно, что дачники – народ странный, почти такой же, как и программисты. Строят они свои дачи непонятно где, да и выращивают там непонятно что и непонятно зачем. А уж как они туда добираются, это другая история: кто на автобусе, кто на электричке, кто на автомобиле, ну а кто-то вовсе пешком ходит от дома и до самого участка. Так что не стоит удивляться, если вдруг Вы узнаете,



что некое садоводческое товарищество располагается на острове, а дачники добираться до него самолетом. Да еще и на этом острове может не быть посадочной полосы, так что высадиться на остров можно, только прыгая с парашютом (мы уж не рассматриваем то, как они возвращаются с дач домой). Рассмотрим этот уникальный случай. Пилот всегда старается осуществить высадку парашютистов таким образом, чтобы дачники приземлялись как можно ближе к своим прямоугольным участкам. Пилоту интересно знать: сколько дачников приземлится на свои участки? Помогите ему решить эту задачу!

## Входные данные

В первой строке входного файла INPUT.TXT записано натуральное число  $N$  ( $1 \leq N \leq 1000$ ) – количество дачников, далее идут  $N$  строк, в каждой из которых описаны координаты каждого дачника и его участка:  
X Y X1 Y1 X2 Y2 X3 Y3 X4 Y4  
где  $(X, Y)$  – координаты приземления парашютиста  
 $(X1, Y1, X2, Y2, X3, Y3, X4, Y4)$  – координаты прямоугольного участка на плоскости, указанные последовательно.  
Все координаты – целые числа, не превышающие 50000 по абсолютной величине

## Выходные данные

В выходной файл OUTPUT.TXT нужно вывести количество дачников, приземлившихся на свой участок. Попадание на границу участка считается попаданием на участок.

## Вариант 34

На плоскости даны две окружности. Требуется проверить, имеют ли они хотя бы одну общую точку.

## Входные данные

Входной файл INPUT.TXT состоит из двух строк. На каждой строке записана информация об одной окружности – координаты ее центра  $x$  и  $y$  (целые числа, по модулю не превосходящие 5000) и радиус (целое число  $1 \leq r \leq 1000$ ).

## Выходные данные

В выходной файл OUTPUT.TXT выведите «YES», если окружности имеют хотя бы одну общую точку, и «NO» в противном случае.

## Вариант 35

Многие из вас, вероятно, знакомы с понятием симметрии относительно прямой. Пусть на плоскости расположена прямая  $L$  и точка  $A$ . Точка  $B$  называется симметричной точке  $A$  относительно прямой  $L$ , если отрезок  $AB$  перпендикулярен прямой  $L$  и делится пополам точкой пересечения с ней. В частности, если точка  $A$  лежит на прямой  $L$ , то точка  $B$  совпадает с точкой  $A$ .

Задана прямая  $L$ , параллельная одной из осей координат, и точка  $A$ . Найдите точку  $B$ , симметричную  $A$  относительно  $L$ .

## Входные данные

Первая строка входного файла INPUT.TXT содержит 4 числа:  $x_1, y_1, x_2, y_2$  – координаты двух различных точек, через которые проходит прямая  $L$ . Вторая

строка входного файла содержит 2 числа  $x_a$  и  $y_a$  – координаты точки А. Все числа во входном файле целые и не превосходят 108 по модулю.

Выходные данные

В выходной файл OUTPUT.TXT выведите числа  $x_b$  и  $y_b$  – координаты точки В.

### Вариант 36

Оператором А, действующим из множества X в множество Y (или просто оператором из X в Y) называется правило, согласно которому каждому элементу x множества X сопоставляется элемент  $y=Ax$  из множества Y. Пусть X и Y – множества точек на плоскости. Оператор А из X в Y называется сжимающим с коэффициентом q, где q – вещественное число из полуинтервала  $[0, 1)$ , если для любого x из X выполнено  $\|Ax\| \leq q \cdot \|x\|$  (здесь  $\|x\|$  - норма точки x – расстояние от x до начала координат). Проще говоря, оператор называется сжимающим с коэффициентом q если он сопоставляет каждой точке точку, которая не менее, чем в q раз ближе к началу координат.

Для заданного оператора А требуется проверить является ли он сжимающим с коэффициентом q.

Входные данные

Первая строка входного файла INPUT.TXT содержит количество точек n ( $1 \leq n \leq 100$ ) и число q ( $0 \leq q < 1$ ), заданное не более чем с 3 знаками после десятичной точки. Следующие n строк содержат по 4 целых числа, по модулю не превосходящих 1000, разделенные пробелами – координаты точки множества X и сопоставленной ей точки из множества Y.

Выходные данные

В выходной файл OUTPUT.TXT выведите одно слово: “Yes” если оператор является сжимающим с коэффициентом q и “No” в противном случае.

### Критерии оценивания

Оценка	Критерии оценивания
Оценка «5» (отлично)	Во введении четко сформулирован тезис, соответствующий теме, выполнена задача заинтересовать слушателя; деление текста на введение, главную часть и заключение; в основной части логично, связно и полно доказывается выдвинутый тезис; заключение содержит выводы, логично вытекающие из содержания основной части; правильно (уместно и достаточно) используются разнообразные средства связи; демонстрирует полное понимание проблемы.

	Все требования, предъявленные к заданию, выполнены.
<b>Оценка «4» (хорошо)</b>	Во введении четко сформулирован тезис, соответствующий теме, в известной мере выполнено задача заинтересовать слушателя; в основной части логично, связно, но недостаточно полно доказывается выдвинутый тезис; заключение содержит выводы, логично вытекающие из содержания основной части; уместно используются разнообразные средства связи; для выражения своих мыслей обучающийся не пользуется упрощенно- примитивным языком.
<b>Оценка «3» (удовлетворительно)</b>	Во введении тезис сформирован нечетко или не вполне соответствует теме; в основной части выдвинутый тезис доказывается недостаточно логично (убедительно) и последовательно; в заключении выводы не полностью соответствуют содержанию основной части; язык работы в целом не соответствует уровню курса, на котором обучается обучающийся.
<b>Оценка «2» (неудовлетворительно)</b>	Во введении тезис отсутствует или не соответствует теме; в основной части нет логичного последовательного раскрытия темы; выводы не вытекают из основной части; отсутствует деление текста на введение, основную часть и заключение; язык работы можно оценить как