**Проект календарно-тематического планирования**

**по теме "Одномерные массивы"**

**учителя информатики**

**МБОУ «Средняя школа № 57»**

**г. Ульяновска**

**Малеева Т.П.**

Оглавление

[Введение 3](#_Toc519529942)

[Роль и место изучения темы «Обработка одномерных массивов» в курсе информатики. 5](#_Toc519529943)

[Методические рекомендации по изучению темы и формированию умений и навыков по обработке большого массива однотипных данных. 6](#_Toc519529944)

[Анализ состояния проблемы 7](#_Toc519529945)

[Заключение 8](#_Toc519529946)

[Библиографический список: 9](#_Toc519529947)

[*Приложение 1* 11](#_Toc519529948)

[*Приложение 2* 13](#_Toc519529949)

Введение

Раздел «Основы алгоритмизации и программирования» имеет явно выраженную практическую направленность, что способствует усилению связи обучения с жизнью. В этом разделе, как ни в одном другом, большое внимание должно уделяться решению задач и выполнению упражнений.

При решении некоторых задач строится математическая модель и вычислительный алгоритм, требующие обоснований, выходящих за рамки школьной программы.

Происходит расширение традиционного, известного из других школьных предметов понятия величина. Вводятся и используются не только числовые, но и литерные величины. Кроме этого, ученики знакомятся с новыми способами организации данных в форме одномерных и двумерных таблиц (массивов).

В образовательный стандарт по информатике входит понятие операций над массивами, в требованиях к знаниям и умениям выпускников прописано, что они должны уметь писать алгоритмы (программы), требующие обработку массивов.

Умение школьников формально исполнять алгоритмы, отслеживать результаты выполнения программы, решать практические задачи с использованием массивов однотипных данных проверяется на ОГЭ по информатике.

Все выше сказанное определяет *актуальность* проекта календарно-тематического плана по теме: «Одномерные массивы».

*Объектом* исследования является педагогический процесс

*Предмет* исследования календарно тематическое планирование

*Целью проекта* является совершенствование методики обучения школьников и осмысление системы задач по обработке одномерных массивов данных на языке программирования Pascal ABC.

*Задачи проекта:*

* рассмотреть роль и место изучения темы «Обработка одномерных массивов» в разделе «Алгоритмизация и программирование» школьного курса информатики.
* обобщить и систематизировать теоретическую информацию по обработке одномерных массивов в среде программирования Pascal ABC;
* предложить методические рекомендации по изучению структурированного типа «массивы» в 9 классе;
* разработать календарно-тематический план по теме: «Одномерные массивы»;
* разработать подборку задач для изучения темы «Обработка одномерных массивов».

*Новизна* исследования заключается в применении в образовательном процессе элективного курса, который является продолжением изучения программы составленной на основе ФКГОС основной школы и разработке методики обучения девятиклассников приемам решения практических задач с использованием одномерных массивов на языке программирования Pascal ABC.

*Практическая значимость проекта* заключается в возможности использовать систему задач для формирования представлений школьников по обработке больших массивов однотипных данных.

Роль и место изучения темы «Обработка одномерных массивов» в курсе информатики.

Одной из задач информатики является нахождение форм представления информации, удобных для компьютерной обработки. Информатика как точная наука работает с формальными (описанными математически строго) структурами данных. К наиболее распространенным структурам данных относятся: целые и вещественные числа, последовательности символов/чисел, строк, таблица (массив), текст, динамические структуры. Структуры данных отражают, в первую очередь, свойства реальных объектов. Используя похожесть структур входных данных и объектов, можно находить эффективные решения задач.

В работе рассматривается одна из сложных структур данных - массив. Необходимость в массивах возникает всякий раз, когда при решении задачи приходится иметь дело с большим, но конечным количеством однотипных упорядоченных данных.

В разделе базового курса «Начало программирования на языке Pascal» структурированные величины, к которым относятся массивы, рассматриваются в ознакомительном порядке. Это обедняет содержание обучения, не дает возможность сформировать целостное представление о понятиях «данные», «обработка информации и данных». Кроме того, устойчивые знания и умения составлять программы для задач практического характера необходимы на экзаменах и олимпиадах по информатике.

Следовательно, без понимания информационной сущности таблиц и основных алгоритмов их обработки невозможно формирование полноценных представлений о возможностях ПК и принципах их работы. Для построения сколь-нибудь сложных и содержательных программ необходимо уверенное владение общими принципами применения таблиц и базовыми приемами их обработки. В приложенном элективном курсе рассмотрены ряд сложных задач, в основе которых лежат знания полученные на уроке.

Методические рекомендации по изучению темы и формированию умений и навыков по обработке большого массива однотипных данных.

1. После знакомства с определением массива необходимо научиться заполнять массив и выводить его элементы на экран дисплея. Поэтому вместе с учащимися необходимо обсудить алгоритмы и написать программы заполнения массива различными способами, используя операторы цикла для обработки каждого элемента массива.

2. На следующем этапе учащиеся разбирают готовые программы и отвечают на поставленный в задании вопрос. Несколько заданий учитель разбирает вместе с учащимися.

3. Следующий этап – знакомство с типовыми алгоритмами обработки массивов и формированием умений использовать эти алгоритмы при решении практических задач. Представленные задачи учитель разбирает вместе с учащимися. Алгоритм решения обсуждается устно, затем оформляется программа.

4. Следующий этап – решение практических задач, требующих умений применять при разработке программы различные типовые алгоритмы. Такие задачи встречаются и в экзаменационных материалах. Алгоритмы решения обсуждаем устно.

Анализ состояния проблемы

В основной школе в 9 классе на изучение информатик и ИКТ отводится 2 часа в неделю (68 часов в год). По авторской программе Л.Л. Босовой на изучение темы начало программирования в Паскаль отведено 16 часов из которых на одномерные массивы – 4 часа. Из выше сказанного понятно, что 4 часа не достаточно на освоение столь значимой темы, так как не хватает времени прорешать многие задачи, которые входят в список заданий ОГЭ, а тем более невозможно подготовить учащихся к олимпиаде по информатике. Но даже если не обращать внимания на такие тонкости, все равно 4 часа очень мало, чтобы подготовить девятиклассников к изучению дальнейших тем, связанных с программированием. Из этого следует несколько путей решения:

1. Увеличение количества часов на изучение данной темы, за счет уплотнения материала курса информатики из других разделов не связанных с программированием. Но это нельзя осуществить без ущерба знаний учащихся.

2. Увеличение количество часов в учебном плане школы, то есть не 2 часа в неделю, а, к примеру, 3 часа. Что тоже довольно проблематично.

3.Вынос углубленного изучения данной темы на факультативные занятия или элективные курсы. На наш взгляд это самый оптимальный вариант, поэтому в приложении предоставлено календарно-тематическое планирование элективного курса по теме: «Одномерные массивы», рассчитанного в количестве 16 часов на продолжение изучения темы во втором полугодии (из расчета один час в неделю). В связи с тем, что тема одномерные массивы изучается в нашей школе в первом полугодии второй четверти, где изучаются ключевые понятия и типовые алгоритмы работы с массивами, данный элективный курс является логическим продолжением и позволяет углублено изучить данный материал. Следует обратить внимание на то, что изучая тему, учащиеся воспринимают данный материал тяжело и без желания, поэтому подробно необходимо изучать только сильным учащимся и тем, кто собирается сдавать ЕГЭ и ОГЭ.

Заключение

Цели проекта календарно-тематического плана по теме: «Одномерные массивы» полностью достигнуты.

Мы рассмотрели роль и место изучения темы «Обработка одномерных массивов» в разделе «Начало программирования в Pascal» школьного курса информатики. В планировании систематизированы теоретические основы по обработке одномерных массивов в среде программирования Pascal ABC. В теоретической части предложены методические рекомендации по изучению структурированного типа «массивы» в 9 классе.

Доступность, логичность, последовательность изложения материала позволяет сильным ученикам познакомиться с ним и отработать все необходимые умения при составлении программ в среде программирования Pascal.

При практической работе каждому из учащихся подбираются задачи по уровню, тем самым с каждым разом усложняя мыслительный процесс.

В проекте предложена оригинальная подборка задач, которые можно использовать для изучения алгоритмов обработки массива и подготовки школьников к ОГЭ.

В заключении хочется отметить, что данная работа поможет многим учителям совершенствовать методику обучения школьников одномерным массивам в языке программирования Pascal и подготовки к ЕГЭ и ОГЭ, а так же к олимпиадам по программированию.

Библиографический список:

1. Информатика и ИКТ: учебник для 9 класса: в 2 ч. Ч. 1 / Л.Л. Босова, А.Ю. Босова. - М.: БИНОМ. Лаборатория знаний, 2012.
2. Информатика и ИКТ. Учебная программа и поурочное планирование для 8-9 классов / Л.Л. Босова, А.Ю. Босова. - М.: БИНОМ. Лаборатория знаний, 2012.
3. Информатика. Задачник - практикум в 2 т. Ч. 1 / Л.А. Залогова, М.А. Плаксин, С.В. русаков и др. под ред. И.Г. Семакина, Е.К. Хеннера: Том 1. - М.: БИНОМ. Лаборатория знаний, 2006.
4. Конспекты уроков информатики в 9-11 классах: практикум по программированию / авт. - сост. А.А. Чернов. - Волгоград: Учитель, 2006.
5. Готовимся к экзамену по информатике / А.Е. Макаренко и др. - М.: Айрис-пресс, 2008.
6. .: Учебник для общеобразовательных учреждений / А.Г. Кушниренко, Г.В. Лебедев, Я.Н. Зайдельман. - М.: Дрофа, 2003.
7. Раздаточные материалы по информатике. 7-9 кл.: В 2 ч. Ч. 2 / Л.А. Анеликова. - М.: Дрофа, 2006.
8. Олимпиадные задания по информатике. 9-11 классы / авт. - сост. Э.С.Ларина. - Волгоград: Учитель, 2007.
9. Информатика. 9-11 классы: подготовка учащихся к олимпиадам. Задачи, упражнения, методические рекомендации / А.В.Мендель, канд. Пед. Наук, Е.М. Волгоград: Учитель, 2009.
10. ФЕДЕРАЛЬНЫЙ ГОСУДАРСТВЕННЫЙ ОБРАЗОВАТЕЛЬНЫЙ СТАНДАРТ СРЕДНЕГО (ПОЛНОГО) ОБЩЕГО ОБРАЗОВАНИЯ [Электронный ресурс]// Режим доступа http://standart.edu.ru/catalog.aspx?CatalogId=6408
11. Методика преподавания информатики: Учеб. пособие для студ. пед. вузов / М.П.Лапчик, И.Г.Семакин, Е.К.Хеннер; Под общей ред. М. П. Лапчика. — М.: Издательский центр «Академия», 2012.
12. Математические основы информатики. Элективный курс: Учебное пособие / Е.В.Андреева, Л.Л.Босова, И.Н.Фалина – М.: БИНОМ. Лаборатория знаний, 2005г.
13. Математические основы информатики. Элективный курс: Методическое пособие / Е.В.Андреева, Л.Л.Босова, И.Н.Фалина – М.: БИНОМ. Лаборатория знаний, 2005г.
14. Информатика и ИКТ. 9 класс. Подготовка к ГИА-2015. Пособие с электронным приложением (CD-диск) / Под ред. Л.Н. Евич, С.Ю.Кулабухова. – Ростов-на-Дону: Легиео, 2014.
15. Сайт Дмитрия Гущина: ОГЭ-2016: информатика. Задачи. Ответы. Обучающая система «РЕШУ ОГЭ» - <http://inf.reshuoge.ru/>

# *Приложение 1*

**КАЛЕНДАРНО-ТЕМАТИЧЕСКОЕ ПЛАНИРОВАНИЕ**

| **№ п/п** | **Дата** | **Тема занятия** | **Краткое содержание** | **Знать / понимать** | **Уметь / применять** | **Практическая часть программы** |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Описание, заполнение и вывод массива | Различные способы описания, заполнения и вывода массива | -заполнение массива с клавиатуры, по закономерности, случайными числами  -вывод массива в строку, в столбец | Вводить и выводить элементы числового одномерного массива | Разработка программы, содержащую заполнение и печать массива |
|  |  | Вычисление суммы, количества и произведения элементов, удовлетворяющих определенным условиям. | Типовые алгоритмы вычисления суммы, количества и произведения элементов, удовлетворяющих определенным условиям. | -алгоритмы нахождения суммы, количества и произведения элементов | Применять алгоритмы нахождения суммы, количества и произведения при решении задач | Разработка программы, содержащую счет элементов массива по условию |
|  |  | Поиск элементов, удовлетворяющих определенным условиям. | Поиск максимального, минимального, отрицательных, положительных, целых, двузначных и д.р элементов массива | -алгоритмы поиска элементов в массиве по заданному правилу | Находить элементы массива по заданному правилу | Разработка программы, содержащую поиск элементов массива по условию |
|  |  | Индексация элементов. | Обработка элементов массива в зависимости от его порядкового номера. | Алгоритмы работы с индексами элементов массива | Находить номера максимального, положительного и д.р. элементов массива | Разработка программы, содержащую перебор элементов массива по индексам |
|  |  | Задачи перестановки. | Алгоритм перестановки элементов местами. Кольцевой сдвиг элементов. | Алгоритм перестановки элементов местами. Кольцевой сдвиг элементов | Применять алгоритм перестановки элементов местами при решении задач | Разработка программы, содержащую перестановки элементов одномерного массива |
|  |  | Сортировка массива. | Способы сортировки | -сортировка методом пузырька;  -сортировка методом прямого выбора | Применять алгоритмы сортировки массива при решении задач | Разработка программы, содержащую сортировку одномерного массива |
|  |  | Задачи, решаемые сортировкой массива. | Решение задач на различные виды сортировок |
|  |  | Выполнение и анализ программы. | Работа с готовыми программами | Умение исполнить циклический алгоритм обработки массива чисел, записанный на алгоритмическом языке | Умение исполнить циклический алгоритм обработки массива чисел | Предвидеть результат выполнения алгоритма и проводить анализ полученного результата |
|  |  | Решение задач из сборника ОГЭ. | Решение задач на обработку массива на примере задания из сборника ОГЕ | Алгоритмы поиска, суммы , произведения, количества, максимального элементов массива | Применять алгоритмы при решении задач | Разработка программы, по условию 20.2 из сборника ОГЭ |
|  |  | Вставка элемента в массив. | Алгоритм вставки элемента в массив, задачи решаемые с помощью алгоритма. | Алгоритм вставки элемента в массив | Применять алгоритм при решении задач | Разработка программы, содержащую вставку элементов одномерного массива |
|  |  | Уплотнение массива. | Алгоритм удаления элемента в массив, задачи решаемые с помощью алгоритма. | Алгоритм удаления элемента в массив | Применять алгоритм при решении задач | Разработка программы, содержащую удаление элементов одномерного массива |
|  |  | Работа с двумя, тремя массивами. | Получение нового массива путем слияния, массива, элементы которого отобраны по специальному правилу. | Алгоритмы обработки нескольких массивов. | Применять алгоритмы при решении задач | Разработка программы, содержащую слияние одномерного массива |
|  |  | Заполнение массива цифрами числа. | Функции обработки строк. Типовой алгоритм: «Разбить» число на цифры, поместив каждую цифру в ячейку массива | Функции, предназначенные для обработки строк: length, copy, val. | Использовать алгоритм разбиения числа на цифры для обработки цифр числа. | Разработка программы, содержащую символьные переменные |
|  |  | Обработка цифр числа. |
|  |  | Решение олимпиадных задач. | Разбор заданий на обработку элементов массива при решении олимпиадных задач. | Типовые алгоритмы обработки одномерных массивов. | Применять несколько типовых алгоритмов при решении олимпиадных задач. | Разработка программы, содержащую несколько видов обработки массива. |
|  |  | Решение олимпиадных задач. |

# *Приложение 2*

**СБОРНИК ЗАДАЧ**

**Занятие 1. Описание, заполнение и вывод массива**

1. Сформировать и вывести массив, каждый элемент которого равен своему индексу.
2. Заполнить массив квадратами натуральных чисел. Вывести на экран элементы через запятую.
3. Сформировать одномерный массив B[1..N], каждый элемент которого вычисляется по формуле B[i]=i2-1.
4. Создайте и распечатайте в строку одномерный массив следующим образом: А[1]=1, A[2]=-1, A[3]=1, A[4]=-1, ... A[n]=-1.
5. Создайте и распечатайте в строку одномерный массив длиной N следующим образом: А[1]=14, A[2]=15, A[3]=16, A[4]=-17, A[5]=-18, А[6]=-19, A[7]=20, A[8]=21, A[9]=22, ... .
6. Написать программу заполнения элементов массива датчиком случайных чисел на интервале: а) [0, 212]; б) [-222, 0]; в) [111, 555]; г) [-50, 100].
7. В математике хорошо известна последовательность Фибоначчи. Она начинается так: 1, 1, 2, 3, 5, 8, 13, 21, ... . первые два члена последовательности равны 1, а каждый следующий равен сумме двух предыдущих. Напишите программу, заполняющий таблицу элементами последовательности Фибоначчи.
8. Дан первый член геометрической прогрессии и ее знаменатель. Найти первые n членов этой прогрессии.

**Занятие 2. Вычисление суммы, количества и произведения элементов, удовлетворяющих определенным условиям.**

1. Подсчитать, сколько раз в массиве встречается элемент, равный заданному с клавиатуры числу.
2. В массиве вещественных чисел определить количество целых.
3. Найти сумму элементов, больших 25, но меньших 50.
4. Найти сумму положительных элементов кратных 7 и 11 одновременно.
5. Найти произведение четных элементов, меньших 5.
6. В массиве натуральных чисел вычислить среднее арифметическое элементов не больших 5.
7. Информация о средней суточной температуре воздуха за месяц задана в виде массива. Определите и выведите на экран, сколько раз температура была ниже среднемесячной.

**Занятие 3. Поиск элементов, удовлетворяющих определенным условиям.**

1. Найти элементы, последняя цифра которых оканчивается на 9.
2. В массиве, элементы которого заданы из промежутка (-1000, 1000) найти все, являющиеся двухзначными числами.
3. Найти разность между минимальным и максимальным элементами массива.
4. Найти максимальный четный элемент массива заданный случайным образом из промежутка [10, 60].
5. Найти максимальный элемент, среди отрицательных элементов массива или вывести на экран сообщение об отсутствии таких элементов.
6. Найти количество элементов равных максимальному.

**Занятие 4. Индексация элементов**.

1. Найти номер максимального элемента и количество элементов, равных максимальному.
2. Вывести на экран четные элементы, кратные 3 находящиеся между 5 и 26 элементами массива А, состоящего из 50 элементов.
3. Сформируйте и выведите на экран массив из девяти элементов каждые три последовательных элемента которого составлены по следующему правилу: 1-й равен текущему индексу, увеличенному в 10 раз; 2-й равен целой части от деления случайного числа на текущий индекс; 3-й равен разности двух предыдущих элементов.
4. Найти элементы, слева и справа от которых стоят четные числа.
5. Найти первый отрицательный элемент массива.
6. Найти первый положительный элемент массива, делящегося на 5 с остатком 2.
7. Найти последний нечетный элемент массива.
8. Дан массив целых чисел, состоящий из 15 элементов. Вывести индексы тех элементов, значения которых больше значения предыдущего элемента.
9. Определить, есть ли 2 пары соседних элементов с одинаковыми знаками.
10. Найти номер последней пары соседних элементов с разными знаками.

**Занятие 5. Задачи перестановки.**

1. В массиве из 25 элементов поменять местами максимальный и минимальный элементы.
2. Переставить элементы массива в обратном порядке.
3. Переставить первый и последний четные элементы массива.
4. Переставить элементы массива, расположенные между первым и последним четными элементами массива в обратном порядке.
5. Переставить элементы массива следующим образом: второй поставить на первое место, третий на второе и т.д. последний на предпоследнее, а первый на последнее.
6. Дан массив целых чисел (n=10), заполненный случайным образом из промежутка

[-40,30]. Переставить первые три и последние три элемента местами, сохраняя их следование.

**Занятие 6,7. Сортировка массива.**

1. Отсортировать массив по убыванию между первым четным и последним нечетным элементами массива.
2. Отсортировать массив по возрастанию между минимальным (включая минимальный) и максимальным (включая максимальный) элементами массива.
3. Дан целочисленный массив из 10 элементов. Вывести на экран все его нечетные элементы, предварительно расположив их по возрастанию методом прямого выбора.
4. Дан целочисленный массив из 10 элементов. Вывести на экран все его отрицательные элементы, предварительно расположив их по убыванию методом пузырька.
5. Заменить все отрицательные элементы на противоположные. Отсортировать полученный массив по убыванию.
6. Дан массив целых чисел, состоящий из 50 элементов заданный случайным образом на промежутке [-100; 100]. Первую половину массива отсортировать по возрастанию, а вторую - по убыванию.

**Занятие 6,7. Выполнение и анализ программы.**

1. В таб­ли­це Dat хра­нят­ся дан­ные из­ме­ре­ний сред­не­су­точ­ной тем­пе­ра­ту­ры за 10 дней в гра­ду­сах (Dat[1] — дан­ные за пер­вый день, Dat[2] — за вто­рой и т. д.). Опре­де­ли­те, какое число будет на­пе­ча­та­но в ре­зуль­та­те ра­бо­ты сле­ду­ю­щей про­грам­мы.

**Var** k, m: **integer**;

Dat: **array**[1..10] **of** **integer**;

**Begin**

Dat[1] := 12;

Dat[2] := 15;

Dat[3] := 17;

Dat[4] := 15;

Dat[5] := 14;

Dat[6] := 12;

Dat[7] := 10;

Dat[8] := 13;

Dat[9] := 14;

Dat[10] := 15;

m := 0;

**for** k := 1 **to** 10 **do**

**if** Dat[k] = 15 **then**

**begin**

m := m+1;

**end**;

**writeln**(m);

**End**.

1. В таб­ли­це Dat хра­нят­ся дан­ные из­ме­ре­ний сред­не­су­точ­ной тем­пе­ра­ту­ры за 10 дней в гра­ду­сах (Dat[1] — дан­ные за пер­вый день, Dat[2] — за вто­рой и т. д.). Опре­де­ли­те, какое число будет на­пе­ча­та­но в ре­зуль­та­те ра­бо­ты сле­ду­ю­щей про­грам­мы.

**Var** k, m: **integer**;

Dat: **array**[1..10] **of** **integer**;

**Begin**

Dat[1] := 12;

Dat[2] := 15;

Dat[3] := 17;

Dat[4] := 15;

Dat[5] := 14;

Dat[6] := 12;

Dat[7] := 10;

Dat[8] := 13;

Dat[9] := 14;

Dat[10] := 15;

m := 0;

**for** k := 1 **to** 10 **do**

**if** Dat[k] > m **then**

**begin**

m := Dat[k]

**end**;

**writeln**(m);

**End**.

1. В таб­ли­це Dat хра­нят­ся дан­ные о ко­ли­че­стве сде­лан­ных за­да­ний уче­ни­ка­ми (Dat[1] за­да­ний сде­лал пер­вый уче­ник, Dat[2] — вто­рой и т. д.). Опре­де­ли­те, какое число будет на­пе­ча­та­но в ре­зуль­та­те ра­бо­ты сле­ду­ю­щей про­грам­мы.

**Var**

k, m: **integer**;

Dat: **array**[1...10] **of** **integer**;

**Begin**

Dat[1] := 7;

Dat[2] := 9;

Dat[3] := 10;

Dat[4] := 5;

Dat[5] := 6;

Dat[6] := 7;

Dat[7] := 9;

Dat[8] := 8;

Dat[9] := 6;

Dat[10] := 9;

m := 10; n: = 0;

**for** k := 1 **to** 10 **do**

**if** Dat[k] < m **then**

**begin**

m := Dat[k];

n := k

**end**;

**writeln**(n);

**End**.

1. В таб­ли­це Tur хра­нят­ся дан­ные о ко­ли­че­стве ребят, хо­див­ших в поход вме­сте с ту­ри­сти­че­ским клу­бом «По­лян­ка». (Tur[1] — число ребят в 2001 году, Tur[2] — в 2002 году и т. д.). Опре­де­ли­те, какое число будет на­пе­ча­та­но в ре­зуль­та­те ра­бо­ты сле­ду­ю­щей про­грам­мы.

**Var**

k, m: **integer**;

Tur: **array**[1..11] **of** **integer**;

**Begin**

Tur[1]:= 1;

Tur[2]:= 11;

Tur[3]:= 8;

Tur[4]:= 12;

Tur[5]:= 5;

Tur[6]:= 6;

Tur[7]:= 15;

Tur[8]:= 16;

Tur[9]:= 16;

Tur[10]:= 21;

Tur[11]:= 7;

m := 0;

**For** k := 1 **to** 11 **Do**

**If** Tur[k] < 10 **Then**

**Begin**

m := m + Tur[k];

**End**;

**Writeln**(m);

**End**.

1. В таб­ли­це Ves хра­нят­ся дан­ные о весе в ки­ло­грам­мах по груп­пе уча­щих­ся спор­тив­ной школы (Ves[1] — вес пер­во­го уча­ще­го­ся, Ves[2] — вес вто­ро­го уча­ще­го­ся и т. д.). Опре­де­ли­те, что будет на­пе­ча­та­но в ре­зуль­та­те вы­пол­не­ния сле­ду­ю­ще­го ал­го­рит­ма

**Var**

i, n: **integer**;

Ves: **array**[1..14] **of** **integer**;

**Begin**

Ves[1]:= 25;

Ves[2]:= 21;

Ves[3]:= 23;

Ves[4]:= 28;

Ves[5]:= 30;

Ves[6]:= 25;

Ves[7]:= 31;

Ves[8]:= 28;

Ves[9]:= 25;

Ves[10]:= 28;

Ves[11]:= 30;

Ves[12]:= 27;

Ves[13]:= 26;

Ves[14]:= 24;

n := Ves[1];

**For** i := 1 **to** 14 **Do**

**If** Ves[i] < n **Then**

**Begin**

n : = Ves[i];

**End**;

**Writeln**(n);

**End**.

1. В про­грам­ме ис­поль­зу­ет­ся од­но­мер­ный це­ло­чис­лен­ный мас­сив A с ин­дек­са­ми от 0 до 9. Зна­че­ния эле­мен­тов равны 5; 1; 6; 7; 8; 8; 7; 7; 6; 9 со­от­вет­ствен­но, т.е. A[0] = 5; A[1] = 1 и т.д. Опре­де­ли­те зна­че­ние пе­ре­мен­ной c после вы­пол­не­ния сле­ду­ю­ще­го фраг­мен­та про­грам­мы

c := 0;

**for** i := 1 **to** 9 **do**

**if** A[i - 1] >= A[i] **then**

**begin**

            t := A[i];

            A[i] := A[i - 1];

            A[i - 1] := t

**end**

**else**

        c := c + 1;

1. В про­грам­ме опи­сан од­но­мер­ный це­ло­чис­лен­ный мас­сив с ин­дек­са­ми от 0 до 10. В при­ве­ден­ном ниже фраг­мен­те про­грам­мы мас­сив сна­ча­ла за­пол­ня­ет­ся, а потом из­ме­ня­ет­ся:

for i : = 0 to 10 do

    A[i] : = i;

for i : = 1 to 11 do

    A[i-1] : = A[11-i];

Чему будут равны эле­мен­ты этого мас­си­ва?

1. В про­грам­ме опи­сан од­но­мер­ный це­ло­чис­лен­ный мас­сив A с ин­дек­са­ми от 0 до 10. Ниже пред­став­лен фраг­мент этой про­грам­мы, в ко­то­ром зна­че­ния эле­мен­тов мас­си­ва сна­ча­ла за­да­ют­ся, а затем ме­ня­ют­ся.

for i:=0 to 10 do

    A[i]:=i-1;

for i:=10 downto 1 do

    A[i-1]:=A[i];

Чему будут равны эле­мен­ты этого мас­си­ва?

**Занятие 9. Решение задач из сборника ОГЭ.**

1. На­пи­ши­те про­грам­му, ко­то­рая в по­сле­до­ва­тель­но­сти на­ту­раль­ных чисел опре­де­ля­ет мак­си­маль­ное число, крат­ное 5. Про­грам­ма по­лу­ча­ет на вход ко­ли­че­ство чисел в по­сле­до­ва­тель­но­сти, а затем сами числа. В по­сле­до­ва­тель­но­сти все­гда име­ет­ся число, крат­ное 5. Ко­ли­че­ство чисел не пре­вы­ша­ет 1000. Введённые числа не пре­вы­ша­ют 30 000. Про­грам­ма долж­на вы­ве­сти одно число — мак­си­маль­ное число, крат­ное 5.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 3 10 25 12 | 25 |

1. На­пи­ши­те про­грам­му, ко­то­рая в по­сле­до­ва­тель­но­сти на­ту­раль­ных чисел опре­де­ля­ет сумму чисел, крат­ных 6. Про­грам­ма по­лу­ча­ет на вход ко­ли­че­ство чисел в по­сле­до­ва­тель­но­сти, а затем сами числа. В по­сле­до­ва­тель­но­сти все­гда име­ет­ся число, крат­ное 6. Ко­ли­че­ство чисел не пре­вы­ша­ет 100. Введённые числа не пре­вы­ша­ют 300. Про­грам­ма долж­на вы­ве­сти одно число — сумму чисел, крат­ных 6.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 3 12 25 6 | 18 |

1. На­пи­ши­те про­грам­му, ко­то­рая в по­сле­до­ва­тель­но­сти на­ту­раль­ных чисел опре­де­ля­ет ко­ли­че­ство чисел, крат­ных 4. Про­грам­ма по­лу­ча­ет на вход ко­ли­че­ство чисел в по­сле­до­ва­тель­но­сти, а затем сами числа. В по­сле­до­ва­тель­но­сти все­гда име­ет­ся число, крат­ное 4. Ко­ли­че­ство чисел не пре­вы­ша­ет 1000. Введённые числа не пре­вы­ша­ют 30 000. Про­грам­ма долж­на вы­ве­сти одно число — ко­ли­че­ство чисел, крат­ных 4.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 3 16 26 24 | 2 |

1. На­пи­ши­те про­грам­му, ко­то­рая в по­сле­до­ва­тель­но­сти на­ту­раль­ных чисел опре­де­ля­ет сумму чисел, окан­чи­ва­ю­щих­ся на 4. Про­грам­ма по­лу­ча­ет на вход ко­ли­че­ство чисел в по­сле­до­ва­тель­но­сти, а затем сами числа. В по­сле­до­ва­тель­но­сти все­гда име­ет­ся число, окан­чи­ва­ю­ще­е­ся на 4. Ко­ли­че­ство чисел не пре­вы­ша­ет 1000. Введённые числа не пре­вы­ша­ют 30 000. Про­грам­ма долж­на вы­ве­сти одно число — сумму чисел, окан­чи­ва­ю­щих­ся на 4.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 3  14  25  24 | 38 |

1. На­пи­ши­те про­грам­му, ко­то­рая в по­сле­до­ва­тель­но­сти на­ту­раль­ных чисел опре­де­ля­ет опре­де­ля­ет сумму всех чисел, крат­ных 6 и окан­чи­ва­ю­щих­ся на 4. Про­грам­ма по­лу­ча­ет на вход на­ту­раль­ные числа, ко­ли­че­ство введённых чисел не­из­вест­но, по­сле­до­ва­тель­ность чисел за­кан­чи­ва­ет­ся чис­лом 0 (0 — при­знак окон­ча­ния ввода, не вхо­дит в по­сле­до­ва­тель­ность). Ко­ли­че­ство чисел не пре­вы­ша­ет 1000. Введённые числа не пре­вы­ша­ют 30 000. Про­грам­ма долж­на вы­ве­сти одно число: сумму всех чисел, крат­ных 6 и окан­чи­ва­ю­щих­ся на 4.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 14 24 144 22 12 0 | 168 |

1. На­пи­ши­те про­грам­му для ре­ше­ния сле­ду­ю­щей за­да­чи. Ка­ме­ра на­блю­де­ния ре­ги­стри­ру­ет в ав­то­ма­ти­че­ском ре­жи­ме ско­рость про­ез­жа­ю­щих мимо неё ав­то­мо­би­лей, округ­ляя зна­че­ния ско­ро­сти до целых чисел. Не­об­хо­ди­мо опре­де­лить мак­си­маль­ную за­ре­ги­стри­ро­ван­ную ско­рость ав­то­мо­би­ля. Если ско­рость хотя бы од­но­го ав­то­мо­би­ля была мень­ше 30 км/ч, вы­ве­ди­те «YES», иначе вы­ве­ди­те «N0».

Про­грам­ма по­лу­ча­ет на вход число про­ехав­ших ав­то­мо­би­лей N (1 < N < 30), затем ука­зы­ва­ют­ся их ско­ро­сти. Зна­че­ние ско­ро­сти не может быть мень­ше 1 и боль­ше 300.Про­грам­ма долж­на сна­ча­ла вы­ве­сти мак­си­маль­ную ско­рость, затем YES или NO.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 4 74 69 63 66 | 74 NO |

1. На­пи­ши­те про­грам­му для ре­ше­ния сле­ду­ю­щей за­да­чи. Ка­ме­ра на­блю­де­ния ре­ги­стри­ру­ет в ав­то­ма­ти­че­ском ре­жи­ме ско­рость про­ез­жа­ю­щих мимо неё ав­то­мо­би­лей, округ­ляя зна­че­ния ско­ро­сти до целых чисел. Не­об­хо­ди­мо опре­де­лить сред­нюю за­ре­ги­стри­ро­ван­ную ско­рость всех ав­то­мо­би­лей. Если не менее двух ав­то­мо­би­лей дви­га­лись со ско­ро­стью не боль­ше 40 км/ч, вы­ве­ди­те «YES», иначе вы­ве­ди­те «N0».

Про­грам­ма по­лу­ча­ет на вход число про­ехав­ших ав­то­мо­би­лей N (1 =< N =< 30), затем ука­зы­ва­ют­ся их ско­ро­сти. Зна­че­ние ско­ро­сти не может быть мень­ше 1 и боль­ше 300.Про­грам­ма долж­на сна­ча­ла вы­ве­сти сред­нюю ско­рость, затем «YES» или «N0».

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 4 74 69 63 96 | 63 NO |

1. На­пи­ши­те про­грам­му для ре­ше­ния сле­ду­ю­щей за­да­чи. Ка­ме­ра на­блю­де­ния ре­ги­стри­ру­ет в ав­то­ма­ти­че­ском ре­жи­ме ско­рость про­ез­жа­ю­щих мимо неё ав­то­мо­би­лей, округ­ляя зна­че­ния ско­ро­сти до целых чисел. Не­об­хо­ди­мо опре­де­лить:

1) раз­ность мак­си­маль­ной и ми­ни­маль­ной ско­ро­стей ав­то­мо­би­лей;

2) ко­ли­че­ство ав­то­мо­би­лей, ско­рость ко­то­рых не пре­вы­ша­ла 30 км/ч.

Про­грам­ма по­лу­ча­ет на вход число про­ехав­ших ав­то­мо­би­лей N (1 < N < 30), затем ука­зы­ва­ют­ся их ско­ро­сти. Зна­че­ние ско­ро­сти не может быть мень­ше 1 и боль­ше 300. Про­грам­ма долж­на сна­ча­ла вы­ве­сти раз­ность мак­си­маль­ной и ми­ни­маль­ной ско­ро­стей ав­то­мо­би­лей, затем ко­ли­че­ство ав­то­мо­би­лей, ско­рость ко­то­рых не пре­вы­ша­ла 30 км/ч.

**При­мер ра­бо­ты про­грам­мы:**

|  |  |
| --- | --- |
| Вход­ные дан­ные | Вы­ход­ные дан­ные |
| 4 74 69 63 96 | 33 0 |

**Занятие 10.** Вставка элемента в массив.

1. После первого элемента, кратного числу 5, вставить элемент, равный 100.
2. После каждого четного элемента массива вставить два элемента с тем же значением.
3. Вставить число *k* перед максимальным (*k* вводить с клавиатуры).
4. Вставить число *k* после всех элементов, кратных своему номеру(*k* вводить с клавиатуры).
5. Дан массив целых чисел (N=10), заполненный случайным образом из промежутка [-40,30]. Вставить число *k* перед всеми элементами, в которых есть цифра 1 (*k* вводить с клавиатуры).

**Занятие 11. Уплотнение массива**.

1. Удалить минимальный из элементов, кратных числу 3.
2. Удалить наибольший из нечетных элементов, расположенных левее минимального из кратных числу 3.
3. Удалить первый элемент массива, у которого последняя цифра четная, а само число делится на нее.
4. Дан массив целых чисел (N=12), заполненный случайным образом трехзначными числами. Удалите последний элемент массива, который состоит из одинаковых цифр.
5. Дан массив целых чисел (N=15).Удалить элементы с 5 по 7.
6. Удалите все элементы оканчивающихся на цифру 5.
7. Удалить все четные элементы массива, имеющие нечетный индекс.

**Занятие 12. Работа с двумя, тремя массивами.**

1. Даны 2 массивы А и В имеют по 15 элементов в каждом. Сформировать массив С, заполненный суммами соответствующих элементов массивов А и В.
2. В конец массива А[n] приписать все элементы массива В[m].
3. Сформировать массив В из отрицательных элементов массива А, массив С из положительных элементов массива А. Массив А не изменять.
4. Подсчитать, сколько элементов массива А совпадают с элементами массива В.
5. Дан одномерный массив, состоящий из Р элементов. Сформировать два новых массива. Элементами первого массива должны быть четные значения элементов, стоящие до первого отрицательного значения элемента исходного массива, а элементами второго массива должны быть нечетные значения элементов, стоящие после первого отрицательного значения исходного массива.
6. Сформулируйте массив, содержащий 15 элементов, с помощью датчика случайных чисел. Получите новый массив, каждый элемент которого равен сумме соответствующего исходного элемента и его квадрата. В полученном массиве найдите максимальный элемент и его номер. Выведите на экран исходные и полученные результаты.
7. Сформулируйте массив, содержащий 15 элементов, с помощью датчика случайных чисел. Получите новый массив, каждый элемент которого равен разности квадрата и самого соответствующего элемента исходного массива. Найдите разность сумм элементов исходного и нового массивов. Выведите на экран исходные и полученные результаты.
8. Сформулируйте массив, содержащий 15 элементов, с помощью датчика случайных чисел. Получите новый массив, каждый элемент которого равен увеличенному на заданное с клавиатуры число соответствующему элементу исходного массивов. Выведите на экран исходные и полученные результаты.

**Занятие 13,14 . Обработка цифр числа.**

1. Найти сумму цифр числа.
2. Найти произведение цифр числа.
3. Найти максимальную цифру числа.
4. Найти количество не нулевых цифр числа.
5. Определить, является ли число, введенное с клавиатуры, числом Армстронга. (число Армстронга: сумма цифр этого числа, возведенных в *n*-ю степень, равна самому числу. Где *n -* количество цифр числа).
6. Найти цифровой корень числа. Цифровой корень числа получается при сложении цифр числа, затем при сложении вновь полученного числа и так до тех пор, пока в сумме не будет получена одна цифра.
7. Найти произведение сверхбольшого числа на цифру.

**Занятие 15,16 . Решение олимпиадных задач.**

**№ 1 «Зарплата сотрудников».**

В фирме «Климат-контроль» 10 сотрудников. В базу данных фирмы заносится информация о зарплате сотрудников.

Напишите программу, которая определяет величину самой большой зарплаты сотрудников и выводит количество сотрудников с этой максимальной зарплатой.

***Формат входных данных:***

С клавиатуры вводятся 10 чисел, обозначающих зарплату 10 сотрудников в рублях.

***Формат выходных данных:***

На экран вывести в виде числа количество сотрудников с максимальной зарплатой.

|  |  |
| --- | --- |
| Ввод | вывод |
| 24 21 23 22 25 24 21 24 25 23 | 2 |

**№ 2 «МОНЕТКИ»**

На столе лежит n монеток. Некоторые из них лежат вверх решкой, а некоторые - гербом. Определите минимальное число монеток, которые нужно перевернуть, чтобы все монетки были повернуты вверх одной и той же стороной.

***Формат входных данных***

В первой строке записано натуральное число N (1≤ N≤100) - число монеток. Во второй строке содержится N целых чисел - 1, если монета лежит вверх решкой и 0, если вверх гербом.

***Формат выходных данных***

На экран вывести число - минимальное количество монет, которые нужно повернуть.

**Пример входных и выходных данных:**

|  |  |
| --- | --- |
| ввод | вывод |
| 5  10110 | 2 |

**№ 3 «ЦИФРОВОЙ ТЕРМОМЕТР»**

Последовательность цифр длиною не менее трех преобразуется следующим образом:

* Для каждой цифры, начиная с третьей, вычисляется сумма предыдущих двух цифр;
* Если последняя цифра суммы больше этой цифры, то она заменяет собой в последовательности эту цифру.

Написать программу, осуществляющую это преобразование.

***Входные данные:*** содержат единственную строчку, в которой записаны цифры исходной последовательности. Количество цифр в последовательности не менее 3 и не более 101000.

***Выходные данные:*** содержат получившуюся в результате последовательность.

**Пример входных и выходных данных:**

|  |  |
| --- | --- |
| вход | 123456328456 |
| выход | 123586428459 |

**№ 4 «СПОРТИВНЫЕ СОРЕВНОВАНИЯ»**

Требуется определить победителей среди участников спортивного соревнования, занявших первые три призовых места по наибольшему количеству очков. Количество участников соревнования не превышает 100 человек. Соревнование построено таким образом, что никакие два или несколько участников не могут набрать одинаковое количество очков. Входная информация представлена в виде массива положительных целочисленных очков участников. В выходной массив требуется занести индексы очков победителей соревнования из исходного массива в порядке возрастания номера их места (первое, второе, третье). Считать, что начальный элемент массива имеет индекс 0.  
***Входные данные:***  
Массив очков участников соревнования X (0 <= Xi <= 100). Элементы массива разделены пробелами. Количество элементов массива от 10 до 100.  
***Выходные данные:***  
Вывести выходной массив. Элементы массива разделить пробелами.  
**Пример входных и выходных данных**

|  |  |
| --- | --- |
| ввод | вывод |
| 20 50 40 80 76 30 75 52 45 70 | 3 4 6 |

**№ 5 «ПЕРЕПРАВА»**

Туристической группе необходимо осуществить переправу через реку. Имеются в наличии две одноместные лодки, связанные канатом, причем весла есть только в одной лодке. Для переправы была придумана следующая схема: 2 туриста переплывают через реку на лодках, затем один из них возвращается с лодками за следующим пассажиром. Время переправы связанных лодок определяется массой наиболее тяжелого участника переправы. Чем больше масса туриста, тем больше времени требуется на переправу. Определить минимальное время переправы для всей туристической группы.

***Вход*** содержит: 1 строка – N (N <= 40) количество участников, последующие N строк содержат время переправы через реку каждого из участников.

***Выход*** должен содержать минимальное время, необходимое группе для переправы через реку.

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| ввод | вывод |
| 3  1  1  5 | 7 |

**№ 6 «БАНКОМАТ»**

В банкомате хранится ограниченное число купюр: K1- достоинством 5000 руб., K2- 1000 руб., K3- 500 руб.,

K4- 100 руб. и K5- 50 руб. Известна сумма S, которую хочет снять пользователь. Найти, сколько и каких купюр он получит (банкомат всегда стремится выдать сначала самые крупные купюры).

***Вход:*** Во входном файле записаны числа K1, K2, K3, K4, K5 (0<=Ki<=100) и S (50<=S<=100000).

***Выход:*** В выходной файл выведите количество 5000, 1000, 500, 100, 50 – рублевых купюр в одной строке через пробел.

|  |  |
| --- | --- |
| **вход** | **выход** |
| 5 5 5 5 5  2850 | 0 2 1 3 1 |
| 1 1 0 7 2 800 | 0 0 0 7 2 |

**№ 6 «ЛЕСТНИЦА».** С крыши дома спущена лестница, состоящая из n ступенек. С каждой ступеньки можно перейти на соседнюю, кроме того, с самой верхней ступеньки можно переступить на крышу, а с самой нижней – на землю. На каждой ступеньке укреплен указатель-стрелка, направленный вверх или вниз. В начальной момент на одной из ступенек стоит человек. В соответствии с указателем он передвигается на соседнюю ступеньку, и сразу после этого указатель меняет свое направление на противоположное. Со следующей ступеньки человек опять переступает в соответствии с указателем-стрелкой, который тут же меняется. Далее человек снова и снова переходит со ступеньки на ступеньку по тому же правилу, пока не окажется на земле или крыше.

Ваша программа должна:

* + - Запросить число ступенек n;
    - Запросить состояние указателей-стрелок каждой из ступенек (снизу вверх) в виде последовательности из n чисел 1(вверх) или -1 (вниз)ж
    - Запросить номер ступеньки k 9нумерация снизу), на которой стоит человек;
    - Определить и сообщить результаты перемещений «на крыше», «на земле», «зациклился».

***Например. Число ступенек 10.***

***Состояние 1, -1,1,-1,-1,1,1,1,1,-1***

***Положение: 5***

***Ответ: «на земле»***

**№ 7 «КЛАВИАТУРА»**

Всем известно, что со временем клавиатура изнашивается, и клавиши на ней начинают залипать. Конечно, некоторое время такую клавиатуру еще можно использовать, но для нажатий клавиш приходиться использовать большую силу.

При изготовлении клавиатуры изначально для каждой клавиши задается количество нажатий, которое она должна выдерживать. Если знать эти величины для используемой клавиатуры, то для определенной последовательности нажатых клавиш можно определить, какие клавиши в процессе их использования сломаются, а какие – нет.

Требуется написать программу, определяющую, какие клавиши сломаются в процессе заданного варианта эксплуатации клавиатуры.

Формат входных данных

Первая строка входного файла содержит целое число n (1 ≤ n ≤ 100) – количество клавиш на клавиатуре. Вторая строка содержит n целых чисел – с1, с2, … , сn, где сi (1 ≤ сi ≤ 100000) – количество нажатий, выдерживаемых i-ой клавишей. Третья строка содержит целое число k (1 ≤ k ≤ 100000) – общее количество нажатий клавиш, и последняя строка содержит k целых чисел pj (1 ≤ pj ≤ n) – последовательность нажатых клавиш.

Формат выходных данных

В выходной файл необходимо вывести n строк, содержащих информацию об исправности клавиш. Если i-ая клавиша сломалась, то i-ая строка должна содержать слово “yes” (без кавычек), если же клавиша работоспособна – слово “no”.

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| вход | выход |
| 5  1 50 3 4 3  16  1 2 3 4 5 1 3 3 4 5 5 5 5 5 4 5 | yes  no  no  no  yes |

**№ 8 «ГАИ»**

Вдоль шоссе в точках X1,X2,...,XN расположены посты ГАИ. В точке X произошло мелкое ДТП (дорожно-транспортное происшествие).

Требуется определить, какой из постов ГАИ расположен ближе всего к этой точке, чтобы с него послать к месту происшествия наряд милиции.

Входные данные

Во входном файле записано сначала число N - количество пунктов ГАИ. (1<=N<=100) Далее следуют координаты расположения постов ГАИ на прямом шоссе (целые числа от -10000 до 10000). Далее идет координата точки, в которой произошло ДТМ (целое число от -10000 до 10000).

Выходные данные

В выходной файл требуется вывести одно число - номер поста ГАИ, с которого нужно послать наряд к месту ДТП. Если несколько постов ГАИ находятся на одинаковом расстоянии от точки ДТП, выведите любой из них.

***Пример входных данных***

5

10 2 8 -7 3

7

***Пример выходных данных***

3

**№ 9 «ЗАДАЧА ИОСИФА ФЛАВИЯ»**

Существует легенда, что Иосиф Флавий - известный историк первого века - выжил и стал известным благодаря математической одаренности. В ходе иудейской войны он в составе отряда из 41 иудейского воина был загнан римлянами в пещеру. Предпочитая самоубийство плену, воины решили выстроиться в круг и последовательно убивать каждого третьего из живых до тех пор, пока не останется ни одного человека. Однако Иосиф наряду с одним из своих единомышленников счел подобный конец бессмысленным - он быстро вычислил спасительные места в порочном круге, на которые поставил себя и своего товарища. И лишь поэтому мы знаем его историю.

В нашем варианте мы начнем с того, что выстроим в круг N человек, пронумерованных числами от 1 до N, и будем исключать каждого k-ого до тех пор, пока не уцелеет только один человек. (Например, если N=10, k=3, то сначала умрет 3-й, потом 6-й, затем 9-й, затем 2-й, затем 7-й, потом 1-й, потом 8-й, за ним - 5-й, и потом 10-й. Таким образом, уцелеет 4-й.)

Задача: определить номер уцелевшего.

***Входные данные***: числа N и k

Ограничения: 1<=N<=500, 1<=k<=100.

***Выходные данные:*** Программа должна выдавать номер уцелевшего человека

|  |  |
| --- | --- |
| вход | выход |
| 10 3 | 4 |

**№ 10 «ПЕРЕСЕЧЕНИЕ МНОЖЕСТВ»**

Даны два неупорядоченных набора целых чисел (может быть, с повторениями). Выдать без повторений в порядке возрастания все те числа, которые встречаются в обоих наборах.

**Входные данные**

В первой строке записано через пробел два целых числа N и M (1 <= N, M <= 106) – количество элементов первого и второго наборов, соответственно. В следующих строках записано сначала N чисел первого набора, а затем M чисел второго набора. Числа разделены пробелами или символами конца строки. Каждое из этих чисел попадает в промежуток от 0 до 105.

**Выходные данные**

Нужно записать в возрастающем порядке без повторений все числа, которые входят как в первый, так и во второй набор. Числа разделять одним пробелом. Если таких чисел нет, то выходной файл должен оставаться пустым.

**Пример**

|  |  |
| --- | --- |
| **вход** | **выход** |
| 11 6  2 4 6 8 10 12 10 8 6 4 2  3 6 9 12 15 18 | 6 12 |

**№ 11 «ЗАКАЗ ИА»**

Кролик открыл в лесу кафе, в котором Сова выпекает очень вкусные пирожные. Ослик Иа решил на свой день рождения угостить друзей и заказал в кафе Кролика множество пирожных самых разных видов в разных количествах. Сова выпекает пирожные «по вдохновению»: то одного вида, то другого, и выкладывает их в один ряд. Кролик пошел на кухню, проверить, сможет ли он выполнить заказ Иа.

Требуется написать программу, определяющую, какие пирожные имеются в нужном количестве, а каких – нет.

***Формат входных данных***

Первая строка содержит целое число n (1 <= n<= 100) – количество видов пирожных, заказанных Иа. Вторая строка содержит n целых чисел – c1, c2 … ,cn, где c1 (1<= c1<=100000) – количество пирожных i-го вида. Третья строка содержит целое число k (1<=k<=100000) – общее количество пирожных на кухне и последняя строка содержит k целых чисел pj (1<= pj<=n) – последовательность выпеченных пирожных.

***Формат выходных данных***

Вывести n строк, содержащих информацию о наличии пирожных. Если пирожных i-го вида достаточно для выполнения, то i-ая строка должна содержать слово “yes” если же пирожных i-го вида не хватает – слово “no”.

**Пример входных и выходных данных**

|  |  |
| --- | --- |
| вход | выход |
| 5  1 5 2 4 3  16  1 2 3 4 5 1 3 3 4 5 5 5 5 5 4 5 | yes  no  yes  no  yes |