

Министерство образования и науки Российской Федерации
Южно-Уральский государственный университет
Кафедра системного программирования

004.4(07)
О-753

ОСНОВЫ ПРОГРАММИРОВАНИЯ

Задания для практических занятий
и методические указания по их выполнению

Челябинск
Издательский центр ЮУрГУ
2013

УДК 004.43(075.8)
О-753

*Одобрено учебно-методической комиссией
факультета вычислительной математики и информатики*

Методическое пособие подготовлено в соответствии с ФГОС ВПО 3-го поколения по образовательному направлению 010300.62 «Фундаментальная информатика и информационные технологии».

*Рецензент:
П.С. Костенецкий*

О-753 **Основы программирования:** Задания для практических занятий и методические указания по их выполнению / сост.: Е.В. Иванова, Т.В. Речкалов, Н.С. Силкина, М.Л. Цымблер. – Челябинск: Издательский центр ЮУрГУ, 2013. – 66 с.

Методическое пособие предназначено для студентов, обучающихся по направлению «Фундаментальная информатика и информационные технологии» при изучении дисциплины «Основы программирования». Пособие содержит тексты задач, контрольные вопросы, указания по выполнению некоторых задач, описание процесса решения задачи, пример оформления практического задания, а также список рекомендуемой литературы [1–8]. Приведены требования и рекомендации по оформлению программ на языке Pascal.

УДК 004.43(075.8)

© Издательский центр ЮУрГУ, 2013

ВВЕДЕНИЕ

Данные методические указания предназначены для выполнения практических работ по дисциплине "Основы программирования" студентами 1 курса направления 010300 "Фундаментальная информатика и информационные технологии".

Структура пособия. Каждая практическая работа содержит *тексты задач* и *контрольные вопросы*, ответы на которые проверяются преподавателем при приеме работы у студента. Текст задачи включает в себя формулировку цели, описание форматов входных и выходных данных и примеры результатов работы программы, решающей данную задачу. Для некоторых задач приводятся указания по их решению.

Каждая задача имеет уникальный в пределах данной лабораторной работы номер. В ссылках на задачу указывается номер задачи и в скобках номер страницы. Например, задача 5 (см. с. 9) означает пятую задачу, описание которой приведено на девятой странице.

Список литературы [1–8] содержит библиографические ссылки на источники, рекомендуемые к прочтению при решении задач.

В приложение вынесены *требования и рекомендации по оформлению исходных текстов программ*, а так же пример оформления практической работы.

Порядок сдачи практической работы. Выполнение студентом практической работы и сдача ее результатов преподавателю происходит следующим образом.

1. Студент выполняет *разработку программ* для всех задач, назначенных ему преподавателем. В том числе разрабатывается *модульная структура*, описываются применяемые алгоритмы, специфицируются и реализуются подпрограммы и тело основной программы. В ходе разработки студент обязан следовать указаниям к данной задаче (в случае их наличия). Исходные тексты программ следует разрабатывать в соответствии с требованиями к оформлению, приведенными в приложении.

В случае затруднений, связанных с разработкой алгоритма, студенту следует обращаться с вопросами к преподавателю, ведущему занятие.

2. Студент выполняет *самостоятельную проверку* исходного текста каждой разработанной программы и правильности ее работы, а также свои знания по теме практической работы.

Исходные тексты программ должны соответствовать требованиям к оформлению, приведенным в приложении. Недопустимо отсутствие в тексте программы следующих важных элементов оформления: спецификации программного файла и подпрограмм, а также отступов, показывающих структурную вложенность языковых конструкций ("лесенка").

Самостоятельная проверка знаний по теме лабораторной работы выполняется с помощью контрольных вопросов и заданий, приведенных в конце текста лабораторной работы.

3. Студент защищает разработанные программы. *Защита* заключается в том, что студент должен ответить на вопросы преподавателя, касающиеся раз-

работанной программы. В том числе вопросы класса, какой будет получен результат в случае использования указанных преподавателем тестовых данных.

К защите необходимо представить исходные тексты программ и сопровождающую информацию по логической организации программы, оформленных в соответствии с требованиями. Пример оформления программы приведен в приложении.

1. ТЕХНИКА РАЗРАБОТКИ ПРОГРАММ

При решении практических работ студент должен получить навык решения задач. Общая схема решения задач для практических работ соответствует жизненному циклу ПО, представленному на рис. 1, за исключением этапа сопровождения.



Рис. 1. Жизненный цикл программного обеспечения

Алгоритм решения задачи для практической работы состоит из следующих этапов:

1. Анализ и спецификация. Студент получает задачу, которую он должен решить. Спецификация – документ, в котором требования задачи записываются на достаточно формальном языке. При решении практических работ условие задачи является спецификацией. Студенту необходимо ознакомиться с условием задачи. На этом этапе студент получает представление о том, что должна делать программная система, и какие ограничения наложены на реализацию программы. Если студенту неясны требования или ограничения для решения задачи, то ему необходимо обратиться к преподавателю для консультации. Если студенту неизвестны знания предметной области задачи, то ему следует обратиться к внешним источникам для получения соответствующих сведений. Результатом этого шага является осознание сути решаемой задачи и понимание формата входных и выходных данных.

2. Проектирование. Проектирование включает в себя разбиение задачи на подзадачи (модули). Модуль автономно решает одну подзадачу. Для каждого модуля необходимо указать спецификацию, включающую в себя назначение модуля, входные и выходные данные. Результатом данного этапа является схе-

ма модульной структуры программы, отражающего связи по управлению между модулями (главный-подчиненный). При составлении модульной структуры следует стремиться к сильной иерархии, при которой один подчиненный модуль имеет один главный модуль. Композиция спроектированных модулей должна обеспечивать решение задачи.

3. Разработка алгоритмов. Этот этап включает в себя описание алгоритмического решения, которое будет использовано в программе. На этом этапе необходимо указать, какие формулы будут использоваться для расчетов (если таковые требуются), какие структуры данных будут использованы и по какой причине (массив, записи, файл, стек и т.д.). Необходимо описать логику работы программных алгоритмов и поведение в частных случаях. Описание алгоритмов выполняется словесно с применением математической нотации и упрощенного псевдоязыка.

4. Кодирование. Кодирование включает в себя создание текстов модулей в виде файлов. Сначала создаются заголовки модулей со спецификациями. Затем создается тело основной программы. Тело основной программы содержит последовательность вызовов модулей, которая приведет к решению задачи. И на последнем шаге происходит реализация тел модулей. Фактически в телах модулей реализуется то, что описано на шаге 3 "Разработка алгоритмов".

5. Компиляция, компоновка. На этом этапе выполняется компиляция программы, подтверждающая ее синтаксическую корректность. Если программа успешно скомпилирована, то можно переходить к следующему шагу, иначе необходимо вернуться на предыдущий.

6. Тестирование. После успешной компиляции выполняется тестовый запуск программы с использованием тестовых данных, для которых заранее известен результат работы программы. В условиях задачи приведены минимальные примечья тестовых данных. Полный набор тестовых данных покрывающий, все основные ситуации работы программы студент должен разработать самостоятельно. Если в результате прохождения теста требуемый результат не совпадает с ожидаемым, необходимо пересмотреть алгоритмическое решение и удостовериться в правильности теста. Если на этом этапе было найдено несоответствие, то необходимо скорректировать тестовый набор данных. Иначе необходимо перейти к шагу отладки. В случае если программа проходит весь набор тестов, то работа над программой считается завершенной.

7. Отладка. На этапе отладке с помощью пошагового исполнения программы выявляются участки кода, не соответствующие описанному алгоритмическому решению. При обнаружении подобного участка необходимо перейти к шагу кодирования и устранить программную ошибку.

Пример оформления практической работы приведен в приложении.

2. ЗАДАНИЯ И МЕТОДИЧЕСКИЕ УКАЗАНИЯ ПО ИХ ВЫПОЛНЕНИЮ

2.1. Простые задачи

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием операторов объявления переменных и присваивания.

1. Вычисление значения выражения

Для заданных вещественных чисел X и Y вычислите значения A , B , C :

$$A = \sqrt[8]{x^8 + y + 8^x}, B = \frac{xy - 3,3 * |x + \sqrt[4]{y}|}{(xy)^3 + \sqrt{|g|x|}}, C = \frac{|\tan x + \sin y^4|}{\cos 2x + |\cot y|}$$

Входные данные: два вещественных числа – X и Y .

Выходные данные: три вещественных числа в формате

$A = \langle \text{значение } \mathbf{A} \rangle$

$B = \langle \text{значение } \mathbf{B} \rangle$

$C = \langle \text{значение } \mathbf{C} \rangle$

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 3	$A = 2.05897$ $B = -0.00006$ $C = 0.44249$
2	3 4	$A = 3.02852$ $B = -0.00003$ $C = 0.62601$

Указания:

1. Реализуйте вычисление значений A , B , C в виде отдельных подпрограмм.
2. Изучите справку по стандартным функциям: Abs , Cos , Exp , Ln , Sin , Sqr , $Sqrt$.

3. Для возведения аргумента $x > 0$ в произвольную дробную степень y , не равную 0.5 (т.е. извлечение квадратного корня), используйте формулу $x^y = \begin{cases} e^{-y \ln x}, & y < 0 \\ e^{y \ln x}, & y \geq 0 \end{cases}$.

4. Для выражения десятичного логарифма через натуральный логарифм используйте формулу $\log x = \frac{\ln x}{\ln 10}$.

5. Для разработки тестов используйте приложение Калькулятор операционной системы Windows. Для использования тригонометрических и других инженерных функций калькулятора в меню Вид этого приложения выберите команду Инженерный.

2. Вычисление длины вектора

Вычислите длину вектора с заданными декартовыми координатами вершин.

Входные данные: четыре вещественных числа – декартовы координаты вершин вектора.

Выходные данные: вещественное число в формате

Длина вектора: <значение длины вектора>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0 0 0 1	Vector's length is 1.00000
2	4 0 0 -3	Vector's length is 5.00000

3. Вычисление периметра и площади треугольника

Вычислите периметр и площадь треугольника с заданными длинами сторон.

Входные данные: три вещественных числа – длины сторон треугольника. Треугольник с заданными длинами сторон заведомо существует.

Выходные данные: два вещественных числа в формате

Периметр: <значение периметра>

Площадь: <значение площади треугольника>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 1 1	Perimeter is 3.00000 Square is 1.00000
2	3 4 5	Perimeter is 12.00000 Square is 6.00000

Указания

Реализуйте вычисление периметра и площади треугольника в виде отдельных подпрограмм.

4. Вычисление периметра и площади треугольника в координатной плоскости

Вычислите периметр и площадь треугольника с заданными декартовыми координатами вершин.

Входные данные: шесть вещественных чисел – декартовы координаты вершин треугольника. Треугольник с заданными вершинами заведомо существует.

Выходные данные: два вещественных числа в формате

Периметр: <значение периметра треугольника>

Площадь: <значение площади>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0 0 0 3 -4 0	Perimeter is 12.00000 Square is 6.00000

Указания:

1. Реализуйте вычисление длины вектора по декартовым координатам его вершин в виде подпрограммы. Используйте решение задачи 2 (с. 7).

2. Реализуйте вычисление периметра и площади треугольника в виде отдельных подпрограмм. Используйте решение задачи 3 (с. 8).

5. Робинзон

Робинзон, будучи на необитаемом острове, считал прожитые дни. Когда корабль забрал Робинзона с необитаемого острова, Капитан корабля должен был записать в судовой журнал, сколько полных лет, месяцев и дней прожил Робинзон на острове. При этом Капитан считал, что в месяце 30 дней, а в году 12 месяцев (т.е. в году 360 дней).

Напишите программу, которая помогает Капитану сделать запись в судовом журнале.

Входные данные: одно целое число D – число дней, которые прожил Робинзон на необитаемом острове, $1 \leq D \leq 36000$.

Выходные данные: вывести, сколько полных лет, месяцев и дней прожил Робинзон на острове, в следующем формате:

Years: <число полных лет>
 Months: <число полных месяцев>
 Days: <число полных дней>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	393	Years: 1 Months: 1 Days: 3
2	1800	Years: 5 Months: 0 Days: 0
3	1050	Years: 2 Months: 11 Days: 0
4	10	Years: 0 Months: 0 Days: 10
5	91	Years: 0 Months: 3 Days: 1
6	36000	Years: 100 Months: 0 Days: 0

Указания:

1. Оформите вычисление количества полных лет, месяцев и дней, прожитых Робинзоном на острове, в виде отдельной подпрограммы.

2. Изучите справку по стандартным операциям целочисленного деления `div` и взятия остатка `mod` и используйте их в реализации данной подпрограммы.

6. Сила притяжения

Масяня бежит навстречу Хрюнделю со скоростью M км/ч. Хрюндель идет навстречу Масяне со скоростью N км/ч. Определить, через какое время встретятся Масяня и Хрюндель, если дорога прямая, а начальное расстояние между ними было S км.

Входные данные: три целых числа: M – скорость, с которой движется Масяня, N – скорость, с которой движется Хрюндель, S – первоначальное расстояние между Масяней и Хрюндедем.

Выходные данные: вывести количество секунд до встречи Масяни и Хрюнделя в следующем формате

<количество секунд> seconds

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	20 5 10	1440 seconds

7. Багаж

Дама сдавала в багаж: диван, чемодан, саквояж, картину, корзину, картонку и маленькую собачонку. При регистрации в аэропорту каждая вещь была взвешена (вес каждой в квитанции указан). Бесплатный провоз – 20 кг багажа и 5 кг ручной клади.

Определить, сколько придется даме доплатить за провоз вещей, если за каждый лишний кг 1 кг багажа нужно заплатить N рублей, за 1 кг ручной клади – M рублей. Корзину, картонку и маленькую собачонку дама берет с собой в самолет в качестве ручной клади.

N , M – константы, считать равными 100 и 150 руб. соответственно.

Входные данные: семь вещественных чисел – вес каждой вещи в соответствии с квитанциями.

Выходные данные: вывести сумму (в рублях), которую необходимо доплатить даме в следующем формате

Необходимо доплатить <количество рублей> рублей

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	53 8 7.2 4 2.4 1.6 0.7	Penalty is 5220 ruble

Указание:

корзину, картонку и маленькую собачонку дама намерена взять с собой в качестве ручной клади.

8. Температура

Известно значение температуры по шкале Цельсия. Найти соответствующие значения температуры по шкалам Фаренгейта и Кельвина. Для пересчета по шкале Фаренгейта нужно исходное значение $t^{\circ}\text{C}$ умножить в 1,8 и прибавить 32, а по шкале Кельвина значение абсолютного нуля соответствует $-273,15^{\circ}\text{C}$.

Входные данные: вещественное число – значение температуры по шкале Цельсия.

Выходные данные: два вещественных числа – значение температуры по шкалам Фаренгейта и Кельвина.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0	32 -273.15

9. Вычисление площади равностороннего треугольника

Дано действительное P – периметр равностороннего треугольника. Составить программу по определению площади данного треугольника.

Входные данные: вещественное число – периметр равностороннего треугольника.

Выходные данные: вещественное число – площадь равностороннего треугольника.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	9	3.89711

10. Возраст ребят

Возраст Коли – X лет, а возраст Миши – Y лет. Найти их средний возраст, а также определить, на сколько отличается возраст каждого мальчика от среднего значения.

Входные данные: два целых числа – возраст мальчиков.

Выходные данные: три вещественных числа:

Average: <средний возраст>

Deviation 1: <отклонение возраста Коли от среднего значения>

Deviation 2: <отклонение возраста Миши от среднего значения>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	10 12	Average: 11.00000 Deviation 1: 1.00000 Deviation 2: 1.00000

11. Полярная система координат

Дана точка в полярной системе координат (R, φ) , где $\varphi \in [0, 360^{\circ})$. Необходимо найти координаты этой точки в декартовой системе координат.

Входные данные: два вещественных числа – точка в полярной системе координат.

Выходные данные: два вещественных числа – точка в декартовой системе координат.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	5 60	2.5 4.33

2.2. Ошибка! Источник ссылки не найден.

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием условного оператора и оператора выбора.

1. Квадратное уравнение

Напишите программу, которая по заданным коэффициентам квадратного уравнения находит его вещественные корни и их количество.

Входные данные: три вещественных числа A, B, C – коэффициенты квадратного уравнения $Ax^2+Bx+C=0$, коэффициент A не может быть равен нулю.

Выходные данные: результаты вычислений выдать в следующем формате:

- если уравнение не имеет вещественных корней
Уравнение
 $(A) * X^2 + (B) * X + (C) = 0$
не имеет вещественных корней
- если уравнение имеет 2 вещественных корня
Уравнение
 $(A) * X^2 + (B) * X + (C) = 0$
имеет 2 вещественных корня:
<1-й корень уравнения (меньший из двух корней)>
<2-й корень уравнения>
- если уравнение имеет 1 вещественный корень
Уравнение
 $(A) * X^2 + (B) * X + (C) = 0$
имеет 1 вещественный корень:
<корень уравнения>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 -7 12	Уравнение $(1.000) * X^2 + (-7.000) * X + (12.000) = 0$ имеет 2 вещественных корня: 3.000 4.000
2	1 4 4	Уравнение $(1.000) * X^2 + (4.000) * X + (4.000) = 0$ имеет 1 вещественный корень: -2.000
3	1 4 5	Уравнение $(1.000) * X^2 + (4.000) * X + (5.000) = 0$ не имеет вещественных корней

Указания:

1. Оформите нахождение вещественных корней и их количества в виде отдельной подпрограммы.
2. Оформите вывод результатов вычислений в виде отдельной подпрограммы.

2. Две даты

Напишите программу, которая по двум заданным датам определяет их взаимное расположение во времени: даты совпадают, первая дата раньше второй или первая дата позже второй.

Входные данные: шесть целых чисел – день, месяц, год первой и второй даты. Даты заведомо корректны.

Выходные данные: выдать одно из следующих сообщений: SAME – если даты совпадают, BEFORE – если первая дата раньше второй даты, AFTER – если первая дата позже второй даты.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 9 2006 1 9 2006	SAME
2	1 9 2006 2 9 2006	BEFORE
3	2 9 2006 1 9 2006	AFTER

3. Вид треугольника

Напишите программу, которая по заданным длинам сторон определяет вид треугольника: треугольник не существует, или треугольник равносторонний, или треугольник равнобедренный, или треугольник общего вида.

Входные данные: три вещественных числа – длины сторон треугольника.

Выходные данные: в зависимости от вида треугольника выдать соответствующее сообщение о виде треугольника.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	-1 2 4	NOT EXISTS
2	0 0 0	NOT EXISTS
3	2 1 1	NOT EXISTS
4	1 1 1	EQUILATERAL
5	2 2 1	ISOSCELES
6	3 4 5	ORDINARY
7	6 7 8	ORDINARY

Указания:

1. Оформите определение, существует ли треугольник с заданными длинами сторон, в виде отдельной подпрограммы.
2. Оформите определение вида треугольника в виде отдельной подпрограммы.

4. Вид треугольника в координатной плоскости

Напишите программу, которая по заданным декартовым координатам вершин треугольника определяет вид этого треугольника: треугольник не существует, или треугольник равносторонний, или треугольник равнобедренный, или треугольник общего вида.

Входные данные: шесть вещественных чисел – декартовы координаты вершин треугольника.

Выходные данные: в зависимости от вида треугольника выдать соответствующее сообщение о виде треугольника.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0 0 0 1 0 2	NOT EXISTS
2	0 0 0 1 1 0	EQUILATERAL
3	-1 0 1 0 0 4	ISOSCELES
4	-1 0 2 0 0 2	ORDINARY

Указания:

1. Оформите в виде отдельной подпрограммы определение факта, лежат ли на одной прямой две точки с заданными декартовыми координатами.
2. Используйте решения задачи 4 (с. 8) и задачи 3 (с. 14).

5. Копейка рубль бережет

Недоверчивый покупатель подсчитывает общую сумму покупок в копейках. Но, подходя к кассе, ему трудно разобраться, сколько рублей и копеек он должен заплатить. Напишите программу, которая поможет ему.

Входные данные: одно целое число из диапазона 1...100 000 – сумма покупки в копейках.

Выходные данные: необходимо выдать запись о том, сколько рублей и копеек должен заплатить покупатель. При этом нужно соблюдать следующие правила:

Слова "рубль" и "копейка" нужно согласовать с числительными. Например: "1 рубль", но "5 рублей", "1 копейка", но "23 копейки".

Вывод числа рублей и копеек нужно производить в отдельных строках. Если число рублей или копеек равно нулю, то соответствующую строку выводить не нужно.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	101	1 РУБЛЬ 1 КОПЕЙКА
2	13	13 КОПЕЕК
3	99999	999 РУБЛЕЙ 99 КОПЕЕК
4	100000	1000 РУБЛЕЙ

Указания:

1. Оформите перевод копеек в рубли и копейки в виде отдельной подпрограммы.
2. Оформите нахождение форм слов "рубль", "копейка", согласованных с соответствующими числительными, в виде отдельных подпрограмм.

6. Рация и чемодан

Штирлиц поручил радистке Кэт подобрать чемодан для перевозки рации. Чемодан и рация имеют форму параллелепипеда. Рацию можно класть в чемодан только вертикально или горизонтально (рацию нельзя ставить под углом к дну чемодана). Боковые стенки рации должны быть параллельны боковым стенкам чемодана. На рис. 2 приведены примеры неправильного и правильного размещения рации в чемодане.

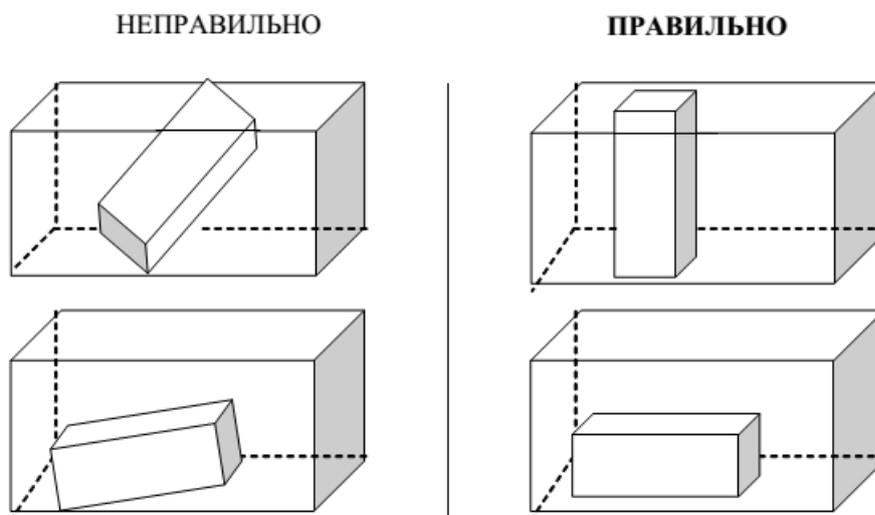


Рис. 2. Примеры неправильного и правильного размещения рации в чемодане

Рация не может поместиться в чемодане, если при этом какой-либо из ее размеров больше или равен соответствующему размеру чемодана.

Напишите программу, которая подскажет радистке Кэт, поместится ли рация в чемодане.

Входные данные: шесть целых положительных чисел – размеры рации и чемодана в сантиметрах.

Выходные данные: сообщение ОК, если рация помещается в чемодане, или NOT ОК в противном случае.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	10 15 5 20 40 30	OK
2	10 15 5 5 5 5	NOT OK
3	10 15 5 15 10 5	NOT OK
4	5 10 40 40 50 10	OK

7. Капитан

Робинзон, будучи на необитаемом острове, считал прожитые дни. Когда корабль забрал Робинзона с необитаемого острова, Капитан корабля должен был записать в судовой журнал, сколько полных лет, месяцев и дней прожил Робинзон на острове. Причем Капитан захотел сделать запись в журнале на русском языке, правильно согласуя числительные и слова "год", "месяц", "день" без ошибок в падежах. При этом Капитан считал, что в месяце 30 дней, а в году 12 месяцев (т.е. в году 360 дней).

Напишите программу, которая поможет Капитану сделать правильную запись в судовом журнале.

Входные данные: одно целое число D – число дней, которые прожил Робинзон на необитаемом острове, $1 \leq D \leq 32000$.

Выходные данные: запись о том, сколько полных лет, месяцев и дней прожил Робинзон на острове. При этом нужно соблюдать следующие правила:

Слова "год", "месяц", "день" нужно согласовать с числительными. Например: "1 год", но "5 лет", "1 месяц", но "3 месяца", "1 день", но "23 дня".

Вывод числа полных лет, месяцев и дней нужно производить в отдельной строке. Если число полных лет, месяцев или дней, прожитых Робинзоном на острове, равно нулю, то соответствующую строку выводить не нужно.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	393	1 ГОД 1 МЕСЯЦ 3 ДНЯ
2	1800	5 ЛЕТ
3	1050	2 ГОДА 11 МЕСЯЦЕВ
4	10	10 ДНЕЙ
5	91	3 МЕСЯЦА 1 ДЕНЬ

Указания:

1. Используйте решение задачи 5 (с. 9).
2. Оформите нахождение форм слов "год", "месяц", "день", согласованных с соответствующими числительными, в виде отдельных подпрограмм.

8. Окружности

Напишите программу, которая определяет взаимное расположение двух окружностей на плоскости.

Входные данные: три целых числа: X –координата центра окружности, Y –координата центра окружности и радиус окружности.

Выходные данные: одно сообщение в соответствии с рис. 3.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0 0 5 0 0 4	ВЛОЖЕНЫ
2	0 0 2 0 0 2	СОВПАДАЮТ
3	2 0 2 0 1 1	ПЕРЕСЕКАЮТСЯ
4	0 0 1 10 -10 1	НЕ ПЕРЕСЕКАЮТСЯ

Указания:

используйте решение задачи 2.

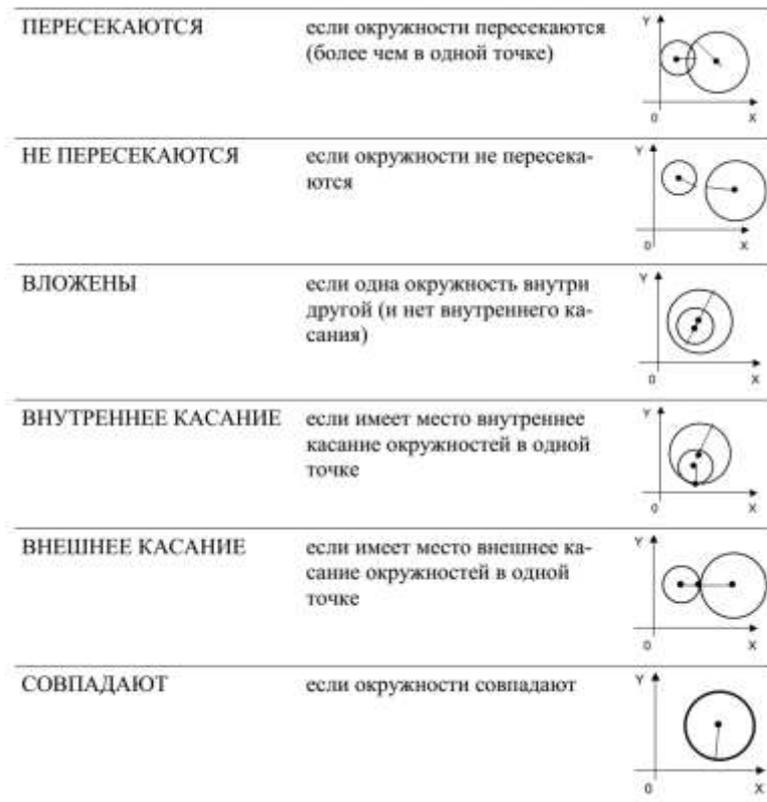


Рис. 3. Виды взаимного расположения двух окружностей

9. Прибытие поездом

Пионер Петечка решил поехать на каникулы к своей любимой бабушке Дусе в деревню, о чем он тут же известил свою бабушку телеграммой. Однако Петечка написал бабушке только время отправления поезда и время пребывания в

пути. Помогите бабушке Дусе вычислить, через сколько дней и в какое время подъедет ее любимый внук, чтобы она смогла его встретить на вокзале.

Входные данные: четыре целых числа – часы и минуты отправления поезда, а также часы и минуты продолжительности поездки.

Выходные данные: результат вычислений в следующем формате:

Peter arrives after <количество дней в пути> days at <время в формате чч:мм>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	22 50 49 30	Peter arrives after 3 days at 00:20

Указание:

за один день принять пребывание Петечки в поезде в 00:00.

10. Шахматы

Координаты поля шахматной доски заданы парой натуральных чисел, не превосходящих восьми. Дано четыре числа k , l и m , n , задающие координаты двух клеток (при счете слева направо и снизу вверх).

Определить:

- А. Одного ли цвета эти два поля?
- В. Если ли на первом поле расположен ферзь, то угрожает ли он фигуре противника, расположенной на втором поле?
- С. Если на первом поле расположен конь, то угрожает ли он фигуре противника, расположенной на втором поле?

Входные данные: четыре целых числа от 1 до 8.

Выходные данные: результат вычислений в следующем формате

- А) Yes/No
- В) Yes/No
- С) Yes/No

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 2 2 4	А) No В) No С) Yes

11. Вид четырехугольника

Даны длины двух смежных сторон четырехугольника и угол между ними. Длины противоположных сторон четырехугольника равны. Напишите программу, которая определяет вид четырехугольника: четырехугольника не существует, квадрат, ромб, параллелограмм.

Входные данные: три вещественных числа – две длины сторон четырехугольника и угол между ними в градусах от 0 до 180.

Выходные данные: в зависимости от вида четырехугольника выдать соответствующее сообщение о виде четырехугольника.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	-1 2 90	NOT EXISTS
2	0 0 0	NOT EXISTS
3	2 1 180	NOT EXISTS
4	1 1 90	SQUARE
5	2 2 30	RHOMB
6	3 4 5	PARALLELOGRAM

Указания:

1. Оформите определение, существует ли четырехугольник с заданными длинами сторон и с заданным углом между ними, в виде отдельной подпрограммы.

2. Оформите определение вида четырехугольника в виде отдельной подпрограммы.

2.3. Операторы цикла

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием оператора цикла.

1. Факториал

Напишите программу, которая находит факториал заданного числа.

Входные данные: одно целое число из диапазона 0...12.

Выходные данные: результат вычислений выдать в следующем формате:
<исходное число>! = <значение факториала>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0	0!=1
2	1	1!=1
3	12	12!=479001600

Указания:

1. Оформите нахождение факториала в виде отдельной подпрограммы.

2. Разработайте четыре различных варианта реализации данной подпрограммы: с операторами цикла for-to, for-downto, while, repeat. Протестируйте все варианты реализации.

2. Все о цифрах в числе

Напишите программу, которая находит количество цифр в десятичной записи целого числа, а также сумму и произведение этих цифр.

Входные данные: одно целое число из диапазона -1 000 000...1 000 000.

Выходные данные: результат вычислений выдать в следующем формате:

NUMBER: <исходное число>

DIGITS: <количество цифр в числе>

SUM: <сумма цифр числа>
 PRODUCT: <произведение цифр числа>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	101	NUMBER: 101 DIGITS: 3 SUM: 2 PRODUCT: 0
2	0	NUMBER: 0 DIGITS: 1 SUM: 0 PRODUCT: 0
3	-99999	NUMBER: -99999 DIGITS: 5 SUM: 45 PRODUCT: 59049

Указания:

оформите нахождение количества цифр, суммы цифр и произведения цифр в виде отдельных подпрограмм. В реализации данных подпрограмм используйте оператор цикла while и оператор цикла repeat.

3. Числа Фибоначчи

Последовательность чисел Фибоначчи определяется с помощью следующей формулы:

$$Fib_n = \begin{cases} 1, n = 0 \\ 1, n = 1 \\ Fib_{n-1} + Fib_{n-2}, n > 1 \end{cases} .$$

Последовательность чисел Фибоначчи имеет вид: 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

Напишите программу, которая для заданного числа n находит n-е число Фибоначчи.

Входные данные: одно целое число из диапазона 0...40.

Выходные данные: результат вычислений выдать в следующем формате:
 Fib(<исходное число n>)=<n-е число Фибоначчи>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0	Fib(0)=1
2	5	Fib(5)=8
3	7	Fib(7)=21
4	40	Fib(40)=165580141

Указания:

оформите нахождение заданного числа Фибоначчи в виде отдельной подпрограммы. В реализации данной подпрограммы используйте оператор цикла for.

4. Гиперфункции

Функции гиперболического синуса \sinh и гиперболического косинуса \cosh определяются следующими формулами: $\sinh x = x + \frac{x^3}{3!} + \frac{x^5}{5!} + \frac{x^7}{7!} + \dots = \sum_{n=0}^{\infty} \frac{x^{2n+1}}{(2n+1)!}$, $\cosh x = 1 + \frac{x^2}{2!} + \frac{x^4}{4!} + \frac{x^6}{6!} + \dots = \sum_{n=0}^{\infty} \frac{x^{2n}}{(2n)!}$.

Напишите программу, которая вычисляет значения этих гиперфункций для заданного числа x с заданной точностью Eps . Считать, что требуемая точность достигнута, если очередное слагаемое по модулю меньше Eps – остальные слагаемые можно не учитывать.

Входные данные: вещественные числа x – аргумент функций и $Eps > 0$ – точность вычислений.

Выходные данные: результат вычислений выдать в следующем формате:

```
Eps is <Eps>:  
sinh(<x>)=<значение функции sinh>  
cosh(<x>)=<значение функции cosh>
```

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	0 0.01	Eps is 0.01000: sinh(0.00000)=0.00000 cosh(0.00000)=1.00000
2	1 0.1	Eps is 0.10000: sinh(1.00000)=1.17500 cosh(1.00000)=1.54167

Указания:

1. Оформите вычисление гиперболических функций от заданного аргумента с заданной точностью в виде отдельных подпрограмм.
2. В реализации данных подпрограммы НЕ используйте решение задачи 1(с. 22).

5. Кока-кола

Представьте себе, что у Вас есть R рублей. Одна бутылка кока-колы стоит B рублей, одну пустую бутылку кока-колы Вы можете сдать в приемный пункт и получить E рублей, а емкость бутылки – C литров.

На все имеющиеся у Вас деньги Вы покупаете кока-колу, выпиваете ее (сами или с друзьями) и сдаете все пустые бутылки в приемный пункт. К полученным деньгам Вы добавляете те деньги, которые у Вас остались от первой покупки, и снова на все деньги покупаете кока-колу. Затем Вы ее снова выпиваете, сдаете все пустые бутылки в приемный пункт и так далее, пока у Вас не останется денег на покупку даже одной бутылки кока-колы.

Напишите программу, которая по заданным числам R , B , E , C определяет, сколько литров кока-колы можно выпить, действуя вышеуказанным способом.

Входные данные: четыре целых числа:

- R – начальное количество рублей, $0 \leq R \leq 32000$;
- B – цена одной полной бутылки кока-колы в рублях, $0 \leq B \leq 32000$;

- E – цена одной пустой бутылки кока-колы в рублях, $0 \leq E \leq 32000$, $E < B$ (иначе кока-колу можно пить вечно!);
- C – емкость одной бутылки кока-колы в литрах, $0 \leq C \leq 32000$.

Выходные данные: одно целое число – количество литров кока-колы, которое можно выпить, действуя вышеуказанным способом.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 2 1 1	2
2	13 3 1 3	18
3	1 2 1 1	0

6. Простое число

Проверить является ли заданное число M простым.

Входные данные: одно целое число.

Выходные данные:

YES – если число M является простым;

NO – если число M не является простым.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2	YES
2	12	NO
3	13	YES

7. Список простых чисел

Напечатать все простые числа, не превосходящие числа M .

Входные данные: одно целое число в диапазоне от 1 до 10000.

Выходные данные: список простых чисел, не превосходящих M , через пробел.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2	1 2
2	12	1 2 3 5 7 11
3	13	1 2 3 5 7 11 13

Указание:

используйте решение задачи 0 (с.26).

8. «AVON»

В организации прямых продаж AVON дистрибьютор каждый месяц находит двух новых партнеров. Те, в свою очередь, продолжают работать точно таким же образом. Через каждые полгода четверть дистрибьюторов прекращают свою деятельность (возможно, что через некоторое время их уговорил другой дистрибьютор вновь начать работать). В городе N взрослых жителей (N вводится с клавиатуры).

Через сколько месяцев более 50% жителей будут заниматься прямыми продажами продукции фирмы AVON, считая, что самый первый дистрибьютор начал работать 3 января 1998 г.?

Входные данные: целое число – количество жителей в городе.

Выходные данные: количество месяцев, которое требуется найти по условию задачи.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	20	3
2	100	4
3	1000	6
4	3280	7

9. Счастливый билетик

Аня верит в приметы и каждый раз, заходя в автобус, хочет заполучить очередной "счастливый" билетик ("счастливым" считается билет с шестизначным номером, у которого сумма первых трех цифр равна сумме последних трех).

Напишите программу, которая поможет Ане подсчитать минимальную сумму, которую она должна потратить на покупку билетов (1 билет стоит 12 руб.), чтобы ей наверняка достался "счастливый" билетик (если у кондуктора заканчивается катушка с билетами, считаем, что она начинает новую, первый билет нумеруется "000001").

Входные данные: одно целое число – номер последнего проданного билета.

Выходные данные: сумма в рублях в следующем формате

<количество рублей> rubles

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	123122	12 rubles
2	123123	108 rubles

10. Нумерация строк программы

Многие знают, что строки программы на языке Бейсик нумеруются, начиная с 10 и далее через десяток. Например, если программа на языке Бейсик состоит из четырех строк, то они будут иметь номера 10, 20, 30, и 40. А задумывались ли Вы, сколько всего цифр потребуется для нумерации строк программы на языке Бейсик? Например, если в программе четыре строки, то на их нумерацию

нужно 8 цифр (четыре нуля и по одной единице, двойке, тройке и четверке). А если в программе миллиард строк?

Напишите программу, которая определяет, сколько всего цифр потребуется для нумерации программы на языке Бейсик, состоящей из заданного количества строк.

Входные данные: целое число из диапазона 1...1 000 000 000 – количество строк в программе на языке Бейсик.

Выходные данные: целое число – общее количество цифр для нумерации строк заданной программы.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	4	8
2	1	2
3	11	24

11. Последняя цифра A^B

Требуется написать программу, которая находит три последних цифры, на которые оканчивается число A^B .

Входные данные: два целых числа из диапазона 1...1 000 000.

Выходные данные: целое число – последние три цифры в числе A^B .

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 7	128
2	1 1000000	1
3	999999 2	1

Указание:

не нужно находить значение числа A^B . Необходимо работать только с последними тремя цифрами во всех операциях расчета.

2.4. Массивы

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием массивов.

1. Чудо-гусли

Плыл как-то купец Садко на своем корабле, вез товары на продажу да на гусях играл. Услыхал ту игру Морской Царь, да так сильно она ему понравилась, что решил Морской Царь себе такие же гусли сладить. Остановил корабль Морской Царь и говорит купцу: "Помоги мне, купец Садко, такие гусли сладить. Есть у меня струны, но не знаю я хитрой тайны, как мне их расставить на гусях, чтоб играли так же волшебю, как у тебя". Отвечал купец Садко Морскому Царю: "Так нужно струны устанавливать: возьми сначала струну, которая дает самый высокий звук и установи ее на левый край гуслей. Потом из оставшихся струн снова возьми струну, которая дает самый высокий звук, и установи ее уже на правый край гуслей. Затем из оставшихся струн опять возьми ту,

которая дает самый высокий звук, и установи ее на левый край гуслей. А потом из оставшихся струн возьми струну с самым высоким звуком, и установи ее на правый край гуслей. И так делай, пока все струны не установишь. Понял?". "Больно мудрено!" – сказал Морской

Царь. – "Сам–то я не справлюсь, придется поручить кому-нибудь".

Напишите программу, которая поможет Морскому Царю установить струны на гуслях.

Входные данные: в первой строке файла входных данных записано целое число N – число струн, $1 \leq N \leq 100$. Далее следует N строк, в каждой из которых записано одно целое положительное число, не превышающее 100, – высота звучания струны с соответствующим номером. Чем больше это число, тем более высокий звук дает струна.

Выходные данные: целые числа – высоты звучания струн, отражающие нужный порядок установки струн на гуслях. Числа следует отделять одним пробелом.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	5 8 4 6 1 3	8 4 1 3 6

2. Автогонки

Перед началом соревнований по кольцевым автогонкам спортсменам выдают карту трассы (см. пример карты на рис. 4). На карте через равные промежутки обозначены спуски, подъемы и ровные участки. Отрицательное число означает спуск, положительное – подъем на соответствующее количество метров, ноль – ровный участок.

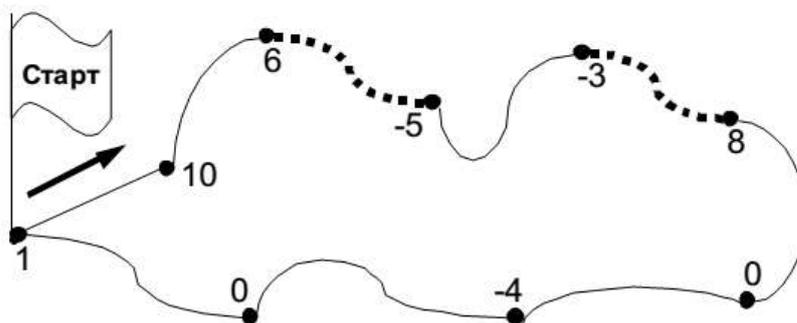


Рис. 4. Пример карты автогонки

Сложным участком трассы называют такой, на котором происходит переход с подъема на спуск или наоборот. На рисунке показан пример трассы, состоящей из 9 участков. Трасса содержит 2 сложных участка, выделенных жирным пунктиром.

Напишите программу, которая по заданной карте находит количество сложных участков трассы.

Входные данные: целое число из диапазона 2...100 – количество участков на карте трассы. Затем соответствующее количество строк с целыми числами из диапазона –100...100 – карта трассы.

Выходные данные: целое число – количество сложных участков трассы.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные	Рисунок
1	9 1 10 6 -5 -3 8 0 -4 0	2	
2	3 0 1 0	0	
3	2 -1 1	2	

3. Шифровка

Агент ФБР Скалли получил задание зашифровать сообщение для агента Малдера. Шифрование выполняется в следующем порядке (пусть шифруется сообщение, состоящее из одного слова АБРАКАДАБРА):

1. Подбирается наименьший квадратный листок клетчатой бумаги, где можно записать сообщение по одному символу в каждой клетке. В нашем случае длина сообщения равна 11, и листок имеет размер 4*4.

2. В каждой клетке листка записывается один символ сообщения в порядке слева направо и сверху вниз. Если некоторые клетки листка остаются пустыми, то они заполняются символом #. В нашем случае листок будет заполнен в соответствии с рис. 5:

А	Б	Р	А
К	А	Д	А
Б	Р	А	#
#	#	#	#

Рис. 5. Пример заполнения шифровального листка

3. Затем символы сообщения записываются с листка по одному в строку в порядке, который указан на рис. 6 стрелками:

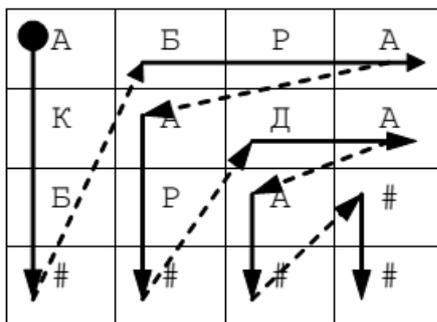


Рис. 6. Порядок обхода шифровального листка

4. То есть сначала записывается 1 столбец, затем остаток 1 строки, далее остаток 2-го столбца, после этого остаток 2 строки и т.д., пока не будут выписаны все символы. В нашем случае зашифрованное сообщение имеет вид АКБ#БРААР#ДАА###.

Напишите программу, которая поможет агенту Скалли зашифровать заданное сообщение.

Входные данные: непустое сообщение, которое нужно зашифровать. Оно состоит только из букв и цифр, его длина не превышает 200. Сообщение может быть записано на нескольких строках, в этом случае символы конца строки и перевода строки шифровать не нужно. Признаком окончания сообщения является точка, которую шифровать не нужно.

Выходные данные: строка зашифрованного сообщения.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	АБРАКАДАБРА.	АКБ#БРААР#ДАА###
2	АБРА КА ДАБРА .	АКБ#БРААР#ДАА###

4. Гарри Поттер и тайная комната

Гарри Поттер ищет тайную комнату в замке Хогвардс. Замок имеет квадратную форму. В каждой комнате замка живет разное количество добрых или злых эльфов.

У Гарри есть карта замка (см. пример карты на рисунке). На карте отмечено количество эльфов в каждой комнате. Если эльфы добрые, то соответствующее число положительно, злые – отрицательно.

Гарри Поттер спросил профессора Дамблдора, как найти тайную комнату. Вот что ответил профессор:

— Пройди каждую строку комнат. Применяй два правила:

1. Если в строке комнат живут только добрые эльфы, то найди ту комнату, в которой живет наименьшее количество добрых эльфов. Иначе в строке комнат найди комнату, в которой живет наибольшее количество злых эльфов.

2. После того, как применил правило 1, смотри все комнаты в том же столбце, что и у найденной комнаты. Если в них живут только злые эльфы, то проверь, живет ли в найденной комнате наименьшее количество злых эльфов. Иначе проверь, живет ли в найденной комнате наибольшее количество добрых эльфов. Если проверка успешна, то это и есть тайная комната. Будь внимателен, так как в замке тайной комнаты может не быть вовсе.

— Ну и ну! Неужели нельзя найти тайную комнату как-то проще?! – подумал Гарри.

На рис. 7 изображена карта замка, имеющего размер 4x4 комнаты. В этом замке есть тайная комната – комната (4,3), в которой живет 5 добрых эльфов.

2	3	4	45
-6	-9	-10	11
-7	-21	-13	12
8	14	5	20

Рис. 7. Пример карты Хогвардса

Напишите программу, которая по заданной карте замка поможет Гарри Поттеру найти тайную комнату и узнать количество эльфов, живущих в ней.

Входные данные: целое число из диапазона 2...20 – количество комнат в строке на карте замка. Затем соответствующее количество строк с целыми числами из диапазона –500...500 – карта замка.

Выходные данные: количество эльфов в тайной комнате или 0 (ноль), если тайной комнаты нет.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	4 2 3 4 45 -6 -9 -10 1 -7 -21 -13 12 8 14 5 20	5
2	2 -10 1 -4 -2	4

5. Больничная статистика

Главному врачу больницы требуются следующие статистические данные о пациентах больницы: минимальная, максимальная и средняя температура. Напишите программу, которая поможет главврачу.

Входные данные: целое число N из диапазона $1...100$ – число пациентов в больнице, и N вещественных неотрицательных чисел, каждое из которых задает температуру пациента.

Выходные данные: результаты вычислений выдать в следующем формате:

Min T is <значение мин температуры с 1 знаком после запятой>

Max T is <значение макс температуры с 1 знаком после запятой>

Avg T is <значение средней температуры с 2 знаками после запятой>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 40.3 36.6 38.7	Min T is 36.6 Max T is 40.3 Avg T is 38.53

Указания:

1. Определите константу "Размерность массива" и тип "Массив целых чисел" соответствующей размерности. Например:

```
Const Size=100;
```

```
Type TArray=array [1..Size] of Integer;
```

2. Оформите нахождение минимального, максимального элементов и среднего арифметического элементов массива в виде отдельных подпрограмм.

6. Матрица

Однажды хакер Томас Андерсон чудом ушел от преследования незнакомцев, которые пытались передать ему какую-то важную информацию. Спасшись, Томас обнаружил, что время ушло далеко вперед, и в реальности люди погружены в глубокий летаргический сон и содержатся в специальных контейнерах, служа источниками жизненной энергии для таинственных электронных монстров. Лишь горстка бодрствующих людей во главе с гуру, которого зовут Морфеус, пытается разрушить зловещий мрак. Чтобы помочь им, Андерсон, который в настоящем мире носит имя Нео, должен правильно складывать и умножать квадратные матрицы. Напишите программу, которая поможет Нео.

Входные данные: целое число N из диапазона $1...10$ – размерность целочисленных матриц A и B . В следующих N строках записаны элементы первой матрицы A .

Далее в N строках записаны элементы матрицы B . В следующей строке записан знак операции: $+$ – необходимо вычислить сумму матриц, $*$ – необходимо вычислить произведение матриц.

Выходные данные: результаты вычислений выдать в следующем формате:

Matrix A is

<элементы матрицы A построчно через пробел>

Matrix B is

<элементы матрицы B построчно через пробел>

Matrix A<знак операции>B is

<элементы результирующей матрицы построчно через пробел >

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 1 0 0 0 1 0 0 0 1 1 2 3 10 30 5 8 23 21 *	Matrix A is 1 0 0 0 1 0 0 0 1 Matrix B is 1 2 3 10 30 5 8 23 21 Matrix A*B is 1 2 3 10 30 5 8 23 21

Указания:

1. Определите константу "Размерность матрицы" и тип "Матрица целых чисел" соответствующей размерности. Например:

```
Const Size=10;
```

```
Type TMatrix=array [1..Size, 1..Size] of Integer;
```

2. Оформите ввод и вывод элементов квадратных матриц в виде отдельных подпрограмм. Обеспечьте ввод элементов матрицы с клавиатуры в виде Введите элемент [..., ...] =>_ (с указанием номера строки и столбца).

3. Оформите нахождение суммы и произведения квадратных матриц в виде отдельных подпрограмм.

7. Супер число

Дана целочисленная матрица размера $M \times N$. Требуется найти: $Y = x(1)x(m) + x(2)x(m-1) + \dots + x(m)x(1)$, где $x(i)$ – разность между максимальным и минимальным элементами i -й строки.

Входные данные: два целых числа $M < 100$ и $N < 100$ – размерности прямоугольной матрицы.

Выходные данные: матрица, размером $M \times N$. Вещественное число Y .

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 3	1 2 3 2 3 4 Y=8

Указания:

1. Определите константу "Размерность матрицы" и тип "Матрица целых чисел" соответствующей размерности. Например:

```
const Size=100;
```

```
type TMatrix=array [1..Size, 1..Size] of Real;
```

2. Ввод матрицы осуществите при помощи оператора Random в отдельной подпрограмме.

3. Оформите вывод элементов матрицы в виде отдельной подпрограммы.

8. «TELE-2»

Компания TELE-2 проводила рекламную акцию. N жителей города получили сертификаты на приобретение телефонных аппаратов по льготным ценам. Во время проведения этой акции разыгрывались призы по 1000 руб. оплаченного времени. Телефонных аппаратов оказалось меньше N и всем счастливицам не хватало. Организаторы решили выдавать аппараты только тем, чьи номера (не более 5-и цифр) на сертификатах имели все различные цифры. Выигрыш в 1000 руб. выдать тем, у кого номера сертификатов являлись простыми числами.

Сколько человек выиграли телефон и сколько человек выиграли призы оплаченного времени?

Входные данные: целое число N – количество выданных сертификатов. Далее следует N строк, каждая из которых содержит номер очередного сертификата, не более 5 цифр.

Выходные данные:

Telephone: <Количество сертификатов, выигравших телефон>

Time: <Количество сертификатов, выигравших 1000 руб. оплаченного времени разговоров>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	10 23122 10200 09929 10100 23456 21426 07219 09859 12312 22222	Telephone: 2 Time: 3

9. Дом Винни-Пуха

Винни-Пух живет в доме квадратной формы. В каждой комнате дома хранится бочонок меда. Готовясь к зиме, Винни-Пух делает обход своего дома. Он заходит в каждую комнату, проверяет, сколько литров меда осталось в бочонке, и записывает результат в свою амбарную книгу. План дома и порядок обхода его комнат изображен на рис. 8. В данном случае на одной стороне дома 4 комнаты, Винни-Пух двигается из комнаты № 1 в комнату № 16 по стрелкам.

Напишите программу, которая помогает Винни-Пуху сделать правильную запись в его амбарной книге.

Входные данные: в первой строке файла входных данных записано одно целое число N – число комнат на одной стороне дома Винни-Пуха, $1 \leq N \leq 10$. Далее идет план дома. Это N строк, в каждой из которых записано N целых неотрицательных чисел, не превосходящих 100. Число показывает количество литров меда в бочонке, который находится в соответствующей комнате дома.

Выходные данные: в файл выходных данных требуется поместить запись об обходе комнат дома. Для каждой посещенной комнаты в отдельной строке следует записать количество литров меда в бочонке, который находится в этой комнате.

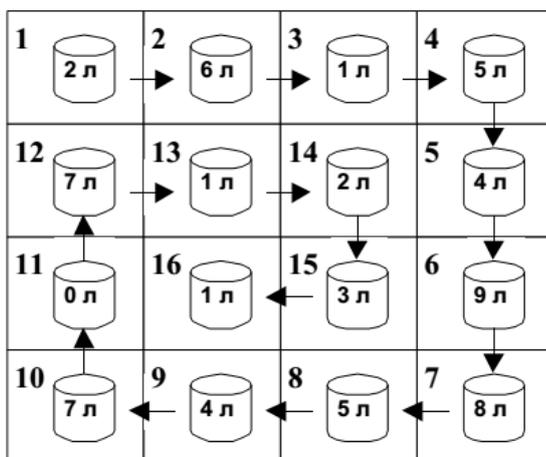


Рис. 8. Пример обхода комнат Винни-Пухом

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 1 2 3 4	1 2 4 3
2	4 2 6 1 5 7 1 2 4 0 1 3 9 7 4 5 8	2 6 1 5 4 9 8 5 4 7 0 7 1 2 3 1

Указания:

задача требует организации обхода матрицы исходных данных "по спирали" и сводится к аккуратному программированию вложенных циклов. При проверке решения следует обратить внимание на граничный случай, когда матрица исходных данных имеет размер 1x1.

10. Транспонированная матрица

Транспонировать квадратную целочисленную матрицу порядка N (поменять местами строки и столбцы матрицы с соответствующими номерами). Вывести на экран исходную и полученную матрицы.

Входные данные: в первой строке файла входных данных записано одно целое число N –порядок матрицы, $1 \leq N \leq 10$. Далее идет матрица. Это N строк, в каждой из которых записано N целых чисел из диапазона $-100 \dots 100$.

Выходные данные: в файл выходных данных требуется поместить запись об обходе комнат дома. Для каждой посещенной комнаты в отдельной строке следует записать количество литров меда в бочонке, который находится в этой комнате.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	4 2 6 1 5 7 1 2 4 0 1 3 9 7 4 5 8	2 7 0 7 6 1 1 4 1 2 3 5 5 4 9 8

11. Дом Винни-Пуха 2

Винни-Пух живет в доме квадратной формы. В каждой комнате дома хранится бочонок меда. Готовясь к зиме, Винни-Пух делает обход своего дома. Он заходит в каждую комнату, проверяет, сколько литров меда осталось в бочонке, и записывает результат в свою амбарную книгу. В какой-то момент Винни-Пух решил, что обходить дом по спирали ему больше не нравится, и он решил сменить способ обхода. Новый план обхода изображен на рис. 9. В данном случае на одной стороне дома 4 комнаты, Винни-Пух двигается из комнаты № 1 в комнату № 16 по стрелкам.

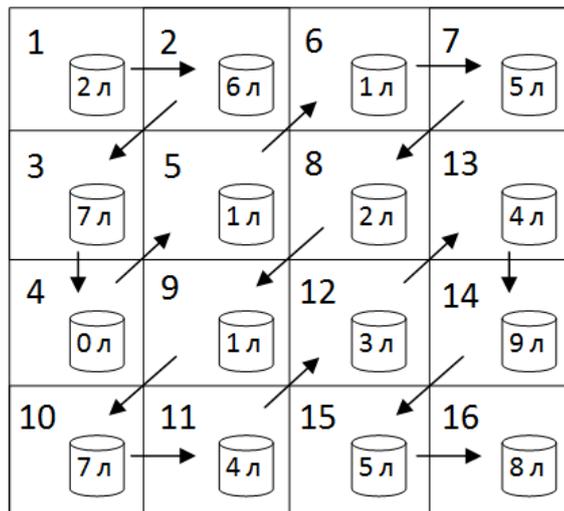


Рис. 9. Новый план обхода комнат Винни-Пухом

Напишите программу, которая помогает Винни-Пуху сделать правильную запись в его амбарной книге.

Входные данные: в первой строке файла входных данных записано одно целое число N – число комнат на одной стороне дома Винни-Пуха, $1 \leq N \leq 10$. Далее идет план дома. Это N строк, в каждой из которых записано N целых неотрицательных чисел, не превосходящих 100. Число показывает количество литров меда в бочонке, который находится в соответствующей комнате дома.

Выходные данные: в файл выходных данных требуется поместить запись об обходе комнат дома. Для каждой посещенной комнаты в отдельной строке следует записать количество литров меда в бочонке, который находится в этой комнате.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 1 2 3 4	1 2 3 4
2	4 2 6 1 5 7 1 2 4 0 1 3 9 7 4 5 8	2 6 7 0 1 1 5 2 1 7 4 3 4 9 5 8

Указания:

задача требует организации обхода матрицы исходных данных "по диагоналям" и сводится к аккуратному программированию вложенных циклов. При проверке решения следует обратить внимание на граничный случай, когда матрица исходных данных имеет размер 1x1.

2.5. Записи

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием записей.

1. Арифметические действия с комплексными числами

Напишите программу, которая находит сумму, разность, произведение, частное и модули двух заданных комплексных чисел.

Входные данные: каждое из комплексных чисел задается значением своей действительной и мнимой частей. Заданные числа отличны от нуля.

Выходные данные: результаты вычислений выдать в следующем формате:

Complex numbers:

A=<число A>

B=<число B>

Results:

|A|=<значение модуля A>

|B|=<значение модуля B>

A+B=<значение A+B>

A-B=<значение A-B>

B-A=<значение B-A>

A*B=<значение A*B>

A/B=<значение A/B>

B/A=<значение B/A>

Формат вывода комплексного числа: (<действительная часть числа>, <мнимая часть числа>), значение действительной и мнимой части числа выводить с 3 знаками после запятой.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 0 0 1	Complex numbers: A=(1.000, 0.000) B=(0.000, 1.000) Results:

№ п/п	Входные данные	Выходные данные
		$ A =1.000$ $ B =1.000$ $A+B=(1.000, 1.000)$ $A-B=(1.000, -1.000)$ $B-A=(-1.000, 1.000)$ $A*B=(0.000, 1.000)$ $A/B=(0.000, -1.000)$ $B/A=(0.000, 1.000)$

Указания:

1. Определите тип данных "Комплексное число". Например:

```
Type TComplex=record
```

```
Re, Im: Real;
```

```
end;
```

2. Оформите ввод и вывод одного комплексного числа, а также математические операции с комплексными числами в виде отдельных подпрограмм.

2. Путешественник Нильс

Нильс очень любит путешествовать и посещает достопримечательности в каждом городе, куда попадает. О каждой достопримечательности Нильс записывает в свою тетрадку название, архитектора, год создания и высоту.

Напишите программу, которая поможет любопытному путешественнику узнать какой из посещенных им памятников самый старый, какой самый высокий, а также распечатывает список памятников, созданных каждым архитектором.

Входные данные: целое число R – количество памятников ($R \leq 50$).

Далее следует информация о каждом из R памятников в следующем формате: <Название> <Архитектор> <Год создания> <Высота>.

Выходные данные: результаты вычислений выдать в следующем формате:

Самый старый памятник: <Название> (<Архитектор>, <Год> г., <Высота> м.);

Самый высокий памятник: <Название> (<Архитектор>, <Год> г., <Высота> м.);

<Архитектор1> (<количество памятников>): <Название1>, ...

<Архитектор2> (<количество памятников>): <Название1>, ...

...

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	7 1: Александровская колонна Монферран О. 1834 47,5 2: Памятник Николаю I Монферран О. 1859 16 3: Исаакиевский собор Монферран О.	Самый старый памятник: Зимний дворец (Растрелли Ф., 1762 г., 30 м.; Самый высокий памятник: Исаакиевский собор (Мон- ферран О., 1858 г., 101,5 м; Монферран О. (3): Алексан-

№ п/п	Входные данные	Выходные данные
	1858 101.5 4: Адмиралтейство Захаров А. 1823 72 5: Казанский собор Воронихин А. 1811 71.5 6: Медный всадник (памятник Петру I) Фальконе Э. 1782 5,3 7: Зимний дворец Растрелли Ф. 1762 30	дровская колонна, Памятник Николаю I, Исаакиевский собор. Захаров А. (1): Адмиралтейство. Воронихин А. (1): Казанский собор. Фальконе Э. (1): Медный всадник (памятник Петру I). Растрелли Ф. (1): Зимний дворец.

3. Адресная книга Вовочки

У Вовочки очень много друзей. Адресная книга Вовочки содержит следующую информацию о его друзьях: фамилии, имена, адреса (город, улица, дом, квартира).

Напишите программу, которая позволит Вовочке найти фамилию и имя друга по заданному адресу, адрес по фамилии друга, а также фамилии тех, кто живет в разных городах, но имеет одинаковый адрес.

Входные данные: целое число R – количество друзей Вовочки ($R \leq 50$).

Далее следует информация о каждом из R друзей Вовочки в следующем формате: <Фамилия> <Имя> <Город> <Улица> <Дом> <Квартира>.

Адрес D в формате: <Город> <Улица> <Дом> <Квартира>.

Фамилия F друга Вовочки.

Выходные данные: результаты вычислений выдать в следующем формате:

По адресу D живет <Фамилия> <Имя>;

F живет по адресу <Город>, <Улица>, <Дом>, <Квартира>;

<Фамилия1> и <Фамилия2> живут по одному адресу, но в разных городах.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 1: Иванов Иван Челябинск пр. Ленина 14 9 2: Сидоров	По адресу г. Челябинск, ул. пр. Ленина, д. 14, кв. 9 живет Иванов; Борисов живет по адресу г. Москва, ул. пр. Ленина, д. 14, кв. 9; Иванов и Борисов живут по одному адресу, но в разных горо-

№ п/п	Входные данные	Выходные данные
	Петр Москва 12 6 3: Борисов Евгений Москва пр. Ленина 14 9 Адрес: Челябинск пр. Ленина 14 9 Фамилия: Борисов	дах.

Указания:

1. Определите тип данных запись "Адрес". Например:

```
Type TAddress=record
  City, Street: String;
  House, Room: Integer;
end;
```

2. Определите тип данных запись "Друг". Например:

```
Type TFriend=record
  FirstName, Surname: String;
  Address: TAddress;
end;
```

3. Оформите ввод и вывод адреса в виде отдельных подпрограмм.

4. Оформите вывод фамилии по заданному адресу, вывод адреса по заданной фамилии, а также вывод фамилий тех, кто живет в разных городах, но имеет одинаковый адрес, в виде отдельных подпрограмм.

5. Организуйте линейный поиск по массиву записей.

4. Ресторан «Шафран»

Хозяин ресторана «Шафран» решил составить готовые предложения для своих посетителей. Меню ресторана содержит следующие сведения о блюдах: название, стоимость, калорийность, к какому виду относится (холодные закуски, первые и вторые блюда, десерт, напитки).

Напишите программу, которая позволит хозяину ресторана составить самое дешевое меню (первое и второе блюда, десерт, напиток), самое дорогое меню (из всех видов блюд) и самое низкокалорийное меню (из всех видов блюд).

Входные данные: целое число R – количество блюд ($R \leq 50$).

Далее следует информация о каждом из R блюд в следующем формате:

<Название> <Вид блюда> <Стоимость> <Калорийность>.

Выходные данные: результаты вычислений выдать в следующем формате:

Самое дешевое меню (<Общая стоимость> руб.): <Название первого блюда>, <Название второго блюда>, <Десерт>, <Напиток>;

Самое дорогое меню (<Общая стоимость> руб.): <Название холодной закуски>, <Название первого блюда>, <Название второго блюда>, <Десерт>, <Напиток>;

Самое низкокалорийное меню (<Общая стоимость> руб.): <Название холодной закуски>, <Название первого блюда>, <Название второго блюда>, <Десерт>, <Напиток>.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	11 1: Рыбное ассорти Холодная закуска 340 1600 2: Сырная палитра Востока Холодная закуска 210 2500 3: Суп Шорпо Первое блюдо 140 2500 4: Борщ Татарский Первое блюдо 160 3000 5: Уха тройная Первое блюдо 200 2600 6: Плов Шади Второе блюдо 210 3000 7: Бешбармак Второе блюдо 250 2700 8: Яблоки в карамели Десерт 150 2000 9: Богатство Насреддина Десерт 110 1000 10: Свежевыжатый сок Напиток 50 300 11: Чай Напиток 25 50	Самое дешевое меню (485 руб.): Суп Шорпо, Плов Шади, Богатство Насреддина, Чай; Самое дорогое меню (990 руб.): Рыбное ассорти, Уха тройная, Бешбармак, Яблоки в карамели, Свежевыжатый сок; Самое низкокалорийное меню (865 руб.): Рыбное ассорти, Суп Шорпо, Бешбармак, Богатство Насреддина, Чай.

Указания:

1. Определите тип данных "Блюдо". Например:

```
Type TDish=record
  Name, Kind: String;
  Price, Kkal: Integer;
end;
```

2. Оформите ввод блюда и вывод меню в виде отдельных подпрограмм.

5. Автосалон

У Ивана Ивановича большой автосалон. Сведения об автомобиле состоят из его марки, номера, года выпуска и фамилии владельца.

Напишите программу, которая позволит Ивану Ивановичу найти владельцев автомобилей заданной марки, владельцев самого старого и самого нового автомобилей, а также количество автомобилей каждой марки.

Входные данные: целое число R – количество автомобилей в салоне ($R \leq 50$).

Далее следует информация о каждом из R автомобилей в следующем формате: <Марка> <Гос. номер> <Год выпуска> <Фамилия владельца>.

Строка D – марка автомобиля.

Выходные данные: результаты вычислений выдать в следующем формате:

Владельцами марки D являются: <Фамилия1>, <Фамилия2>, ...;

Владельцем самого старого автомобиля марки <Марка>, <Гос. номер> является <Фамилия>.

Владельцем самого нового автомобиля марки <Марка>, <Гос. номер> является <Фамилия>.

Количество автомобилей марки <Марка1> – <количество автомобилей>).

Количество автомобилей марки <Марка2> – <количество автомобилей>).

...

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	4 1: Toyota Land Cruiser X123KC 2008 Иванов 2: Nissan Evolution M345MB 2007 Петров 3: ВАЗ 2106 A777AA 1994 Сидоров 4: Toyota Land Cruiser B999BA 2004 Сидоров Toyota Land Cruiser	Владельцами марки Toyota Land Cruiser являются: Иванов, Сидоров; Владельцем самого старого автомобиля марки ВАЗ 2106, A777AA является Сидоров. Владельцем самого нового автомобиля марки Toyota Land Cruiser, X123KC является Иванов. Количество автомобилей марки Toyota Land Cruiser – 2. Количество автомобилей марки Nissan Evolution – 1. Количество автомобилей марки ВАЗ 2106 – 1.

Указания:

1. Определите тип данных "Автомобиль". Например:

```
Type TAuto=record  
Name, GNum, Owner: String;  
Year: Integer;  
end;
```

2. Оформите ввод информации по автомобилям, вывод подсчет количества автомобилей некоторой марки, вывод информации по автомобилю в виде отдельных подпрограмм.

6. Соревнование по плаванию

На соревновании по плаванию приняло участие R команд по 5 человек в каждой команде. Про каждого участника соревнования известно время, за которое он проплыл дистанцию, и его фамилия.

Напишите программу, которая поможет жюри определить команду-победитель по наименьшему суммарному времени преодоления дистанции, а также определить абсолютного победителя соревнования.

Входные данные: целое число R – количество команд ($R \leq 10$).

Далее следует информация о каждом из участников каждой из R команд в следующем формате: <Фамилия> <Время>.

Выходные данные: результаты вычислений выдать в следующем формате:

Команда-победитель: <Фамилия1>, <Фамилия2>, ..., <Фамилия5>.

Абсолютный победитель соревнования: <Фамилия>.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 1: Иванов 20 Петров 22 Сидоров 18 Егоров 21 Андреев 22 2: Александров 19 Бородин 20 Воронин 16 Кузнецов 17 Лешин 18	Команда-победитель: Александров, Бородин, Воронин, Кузнецов, Лешин. Абсолютный победитель соревнования: Воронин.

Указания:

1. Определите тип данных "Участник". Например:

```
Type TPerson=record
  Name: String;
  Result: Integer;
end;
```

2. Определите тип данных "Команда". Например:

```
Type TTeam=record
  Num: Integer;
  Players: array [1..5] of TPerson;
end;
```

3. Оформите ввод информации по каждому участнику и каждой команде в виде отдельных подпрограмм.

7. Радиостанция «Волна»

На радиостанции «Волна» имеется R музыкальных CD-дисков. Про каждый из дисков известно количество (до 20) записанных музыкальных произведений, название и продолжительность каждой композиции, а также жанр каждой композиции (рок, романс, песня).

Напишите программу, которая позволит ди-джею на радио определить жанр, название и продолжительность самой длинной и самой короткой музыкальной композиции, а также список музыкальных композиций заданного жанра.

Входные данные: целое число R – количество CD-дисков ($R \leq 50$).

Далее следует информация о каждом из R дисков в следующем формате:
<Количество композиций>

<Жанр1> <Продолжительность1> <Название1>

<Жанр2> <Продолжительность2> <Название2>...

Строка F – жанр композиции для поиска.

Выходные данные: результаты вычислений выдать в следующем формате:

Самая длинная композиция: <Жанр>-<Название>-<Продолжительность>.

Самая короткая композиция: <Жанр>-<Название>-<Продолжительность>.

Композиции жанра F : <Название1>, <Название2>,...

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 5 Рок 4,5 Stairway to Heaven Рок 3,8 Freebird Рок 4,8 Bohemian Rhapsody Рок 5,2 Smoke on the Water Рок 4,8 Paranoid 4 Песня 3,4 Зайка моя Песня 3,2 Настоящий полковник Рок 3,3 Whole Lotta Love Романс 5,0 Очи черные Рок	Самая длинная композиция: Рок- Smoke on the Water- 5,2 Самая короткая композиция: Песня-Настоящий полковник- 3,2. Композиции жанра Рок: Stairway to Heaven, Freebird, Bohemian Rhapsody, Smoke on the Water, Paranoid, Whole Lotta Love.

8. Клуб любителей кошек «Лемур»

Городской клуб любителей кошек «Лемур» проводил выставку кошек. На каждого участника выставки была заполнена анкета (предполагается, что каждый участник выставки является единственным питомцем хозяина), которая содержит следующую информацию о кошке: кличка, порода, возраст и фамилия хозяина кошки.

Напишите программу, которая позволит организаторам выставки определить какие породы были представлены на выставке, какого возраста кошек было больше всего, встречались ли повторяющиеся клички кошек и если встречались, то какие именно.

Входные данные: целое число R – количество участников выставки ($R \leq 50$).

Далее следует информация о каждом из R участников в следующем формате: <Кличка> <Возраст> <Фамилия хозяина> <Порода>

Выходные данные: результаты вычислений выдать в следующем формате:

На выставке были представлены: <Порода1>-<Порода2>-....

На выставке больше кошек было возраста: <Возраст>.

Если повторяющихся кличек не было:

Повторяющихся кличек не было.

Если повторяющиеся клички были:

Повторяющиеся клички: <Кличка1>, <Кличка2>,....

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 Мурзик 2 Иванов Сибирская Тузик 1,5 Сидоров Русская голубая Мишель 1,5 Петров Сибирская	На выставке были представлены: Сибирская-Русская голубая. На выставке больше кошек было возраста: 1,5. Повторяющихся кличек не было.
2	4 Мурзик 2 Иванов Сибирская Тузик 1,5 Сидоров Русская голубая Мишель 2,5 Петров Сибирская Мурзик 2,5 Андреев Британская короткошерстная	На выставке были представлены: Сибирская-Русская голубая-Британская короткошерстная. На выставке больше кошек было возраста: 2,5. Повторяющиеся клички: Мурзик.

9. Хоккейный турнир

Всего на турнире участвовало R хоккейных команд. Про игроков каждой хоккейной команды известно: фамилия, возраст, роль в команде (капитан, вратарь, 1 нападающий, 2 нападающий, 1 защитник, 2 защитник), а также количество забитых шайб.

Напишите программу, которая поможет жюри определить самого результативного нападающего (больше всех забившего шайб), самого молодого вратаря, самого старшего по возрасту среди капитанов, а также определить средний возраст игроков в каждой из команд.

Входные данные: целое число R – количество команд ($R \leq 10$).

Далее следует информация о каждом из участников каждой из R команд в следующем формате: <Фамилия> <Возраст> <Роль в команде> <Количество забитых шайб>.

Выходные данные: результаты вычислений выдать в следующем формате:

Самый результативный нападающий: <Фамилия>.

Самый молодой вратарь: <Фамилия>.

Самый старший капитан: <Фамилия>.

Средний возраст 1 команды: <Средний возраст1>.

Средний возраст 2 команды: <Средний возраст2>,....

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 1: Иванов 23 капитан 3 Петров 22 нападающий1 3 Сидоров 18 нападающий2 3 Егоров 21 защитник1 0 Андреев 22 защитник2 1 Алехин 20 вратарь 0 2: Александров 19 капитан 5 Бородин 19 нападающий1 2 Воронин 17 нападающий2 4 Кузнецов 17 защитник1 1 Лешин 18 защитник2 1 Ванин 21 вратарь 0	Самый результативный нападающий: Воронин. Самый молодой вратарь: Алехин. Самый старший капитан: Иванов. Средний возраст 1 команды: 21, Средний возраст 2 команды: 18,5.

10. Туристическое агентство «Золотой ключик»

Туристическое агентство «Золотой ключик» предлагает большой ассортимент экскурсионных автобусных туров. По каждому туру известна следующая информация: стоимость тура, продолжительность тура, количество городов (не больше 10), список городов, входящих в экскурсионную программу тура.

Напишите программу, которая поможет будущему туристу определить экскурсионный тур, включающий наибольшее количество городов (насыщенный тур), наиболее дешевый тур, а также туры, включающие интересующий его город.

Входные данные: целое число R – количество экскурсионных туров ($R \leq 10$).

Далее следует информация о каждом из R туров в следующем формате:

<Название> <Стоимость> <Количество городов> <Город1>-<Город2>-...

Строка F – название интересующего города.

Выходные данные: результаты вычислений выдать в следующем формате:

Наиболее насыщенный тур: <Название> (<Стоимость> руб.).

Наиболее дешевый тур: <Название> (<Стоимость> руб.).

Город F можно посетить с турами: <Название1>, <Название2>,...

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 Золотое кольцо 11250 8 Сергиев Посад – Переславль – Ростов Великий –Кострома – Ярославль – Боголюбovo – Суздаль – Владимир Италия в миниатюре 27250 16	Наиболее насыщенный тур: Италия в миниатюре (27250 руб.). Наиболее дешевый тур: Золотое кольцо (11250 руб.). Город Суздаль можно посетить с турами: Золотое кольцо.

№ п/п	Входные данные	Выходные данные
	Брест - Вена - Венеция - Флоренция - Рим - Пиза - Генуя - Ницца - Канны - Грас - Монако - Монте - Карло - Верона - Мюнхен - Брест Суздаль	

11. Чеки

Имеются N чеков из магазина. Для каждого товара указаны следующие характеристики: название, категория (продукты, канцелярия, одежда, бытовое), цена без скидки, скидка.

Напишите программу, которая поможет найти самый дорогой чек, самый дорогой купленный товар по цене с учетом скидки в каждом из чеков, цену среднего чека и наиболее затратную категорию товаров.

Входные данные: целое число N – количество чеков ($N \leq 10$).

Далее следует число M – количество товаров в покупке ($M \leq 5$).

Далее следует информация о каждой из покупок в следующем формате: <название> <категория> <цена без скидки> <скидка>. Цена представлена целым числом и указана в копейках. Скидка представлена в виде процента от цены и указана в виде целого числа от 0 до 100.

Выходные данные: результаты вычислений выдать в следующем формате:

Самый дорогой чек: <номер> <цена> руб.

Цена среднего чека: <цена> руб.

Наиболее затратная категория товаров: <категория> <цена> руб.

Самый дорогой товар в чеке 1: <название> <цена> руб.,

Самый дорогой товар в чеке 2: <название> <цена> руб.,...

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	2 3 Молоко продукты 5100 5 Йогурт продукты 1400 3 кружка бытовое 7000 0 4 Куртка одежда 150000 10 Тапочки одежда 15000 100 Бритва бытовое 70000 5 Тетрадь канцелярия 2500 0	Самый дорогой чек: 2 2040.00 руб. Цена среднего чека: 1086,01 руб. Наиболее затратная категория товаров: одежда 2040.00 руб. Самый дорогой товар в чеке 1: кружка 70.00 руб., Самый дорогой товар в чеке 2: куртка 2040.00 руб.

Указания:

1. Проводите все расчеты в копейках с использованием целочисленных типов данных.

2. Создайте отдельную функцию для форматированного представления денежной суммы в рублях.

2.6. Текстовые файлы

Разработайте алгоритм решения задачи и реализуйте его в виде программы на языке высокого уровня с использованием текстовых файлов.

1. Статистика символов текста

Наборщику текста в типографии необходимо знать, сколько в тексте статьи букв, цифр и знаков препинания. Напишите программу, которая поможет ему.

Входные данные: текст, заканчивающийся символом косой черты / в отдельной строке, расположенный в файле INPUT.TXT.

Выходные данные: необходимо выдать запись в файл OUTPUT.TXT о том, сколько в тексте статьи букв, цифр и знаков препинания. При этом

<буква>::=A | B | ... | Z | a | b | ... | z

<цифра>::=0 | 1 | ... | 9

<знак препинания>::=. | , | ? | !

Формат вывода:

Text contains

<количество букв> letters

<количество цифр> digits

<количество знаков препинания> punctuation marks

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	To be or not to be? That is a question. Быть или не быть? Вот в чем вопрос. /	Text contains 28 letters 0 digits 4 punctuation marks
2	101 /	Text contains 0 letters 3 digits 0 punctuation marks
3	/	Text contains 0 letters 0 digits 0 punctuation marks

2. Простое кодирование текста

Один из самых простых способов закодировать текст заключается в следующем. Выбирается целое число – сдвиг кодирования. Закодированная буква получается из исходной буквы путем циклического сдвига в алфавите на заданное число позиций. Если сдвиг кодирования отрицательный, то циклический сдвиг осуществляется влево, если положительный, то вправо, если сдвиг равен нулю, то кодирования не происходит. Например, если сдвиг кодирования равен -1, то строка АБРАКАДАБРА будет закодирована как ЯАПЯЙЯГЯАПЯ. Напишите программу, которая кодирует текст указанным способом.

Входные данные: в первой строке входных данных записано целое число из диапазона -64 000...64 000 – сдвиг кодирования. В последующих строках записан непустой текст, предназначенный для кодирования. Текст заканчивается

символом кривой черты / в отдельной строке, который не подлежит обработке. Текст записан в файле INPUT.TXT.

Выходные данные: записать закодированный текст в файл OUTPUT.TXT. В закодированном тексте должны быть изменены только буквы латиницы и кириллицы, а пробелы, знаки пунктуации и прочие символы должны остаться без изменений. Символы Ё, Й, Ъ, Ь считаются буквами кириллицы. Регистры соответствующих символов закодированного и исходного текста должны совпадать.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 To be or not to be? АБРАКАДАБРА /	Up cf pq opu up cf? ВВСЬЛВЕВВСВ
2	-1 To be or not to be? АБРАКАДАБРА /	Sn ad nq mns sn ad? ЯАПЯЙГЯАПЯ
3	0 To be or not to be? АБРАКАДАБРА /	To be or not to be? АБРАКАДАБРА

Указания:

1. Оформите кодирование русских и латинских букв в виде отдельных функций.

2. При реализации функции кодирования русских букв учтите, что, в отличие от латинских букв, русские буквы в таблице кодов ASCII занимают два отдельных промежутка. Выясните величину разрыва между этими промежутками с помощью стандартных функций Chr и Ord.

3. Простое декодирование текста

Один из самых простых способов закодировать текст заключается в следующем. Выбирается целое число – сдвиг кодирования. Закодированная буква получается из исходной буквы путем циклического сдвига в алфавите на заданное число позиций. Если сдвиг кодирования отрицательный, то циклический сдвиг осуществляется влево, если положительный, то вправо, если сдвиг равен нулю, то кодирования не происходит. Например, если сдвиг кодирования равен -1, то строка АБРАКАДАБРА будет закодирована как ЯАПЯЙГЯАПЯ. Напишите программу, которая декодирует текст, закодированный указанным способом.

Входные данные: в первой строке входных данных записано целое число из диапазона -64000...64000 – сдвиг кодирования. В последующих строках записан непустой текст, предназначенный для декодирования. Текст заканчивается символом кривой черты / в отдельной строке, который не подлежит обработке. Текст записан в файле INPUT.TXT.

Выходные данные: записать декодированный текст в файл OUTPUT.TXT. В декодированном тексте должны быть изменены только буквы латиницы и кириллицы, а пробелы, знаки пунктуации и прочие символы должны остаться без изменений. Символы Ё, Й, Ъ, Ь считаются буквами кириллицы. Регистры соответствующих символов декодированного и исходного текста должны совпадать.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	1 Up cf pq opu up cf? ВВСВЛБЕЕВВСВ /	To be or not to be? АБРАКАДАБРА
2	-1 Sn ad nq mns sn ad? ЯАПЯЙГЯАПЯ /	To be or not to be? АБРАКАДАБРА
3	0 To be or not to be? АБРАКАДАБРА /	To be or not to be? АБРАКАДАБРА

Указания:

1. Оформите декодирование русских и латинских букв в виде отдельных функций.
2. При реализации функции декодирования русских букв учтите, что, в отличие от латинских букв, русские буквы в таблице кодов ASCII занимают два отдельных промежутка. Выясните величину разрыва между этими промежутками с помощью стандартных функций Chr и Ord.

4. Любимые уроки

Текстовый файл содержит данные опроса учащихся о любимых предметах. Каждый ученик записывал в строку только один предмет. Какой предмет самый любимый учениками данной школы? Какие предметы встретились в списке только один раз?

Входные данные: в каждой строке входных данных записан один предмет. Текст записан в файле INPUT.TXT.

Выходные данные: необходимо выдать запись в файл OUTPUT.TXT о любимых предметах и предметах, встречающихся только один раз.

Формат вывода:

Favourite:

<любимый предмет>

Only:

<предметы, встречающиеся один раз>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Русский язык Математика Английский язык Математика	Favourite: Математика Only: Русский язык Английский язык

5. Повторяющийся символ

В литерном файле найти символ, повторяющийся максимальное число раз. Напечатать этот символ и количество его повторений.

Входные данные: в каждой строке входных данных записан один предмет. Текст записан в файле INPUT.TXT.

Выходные данные: необходимо выдать в файл OUTPUT.TXT символ, повторяющийся максимальное число раз и через пробел число его повторений.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Двадцать первое. Ночь. Понедельник.	e 4

6. Зимняя сессия

Во входном файле содержится информация об итогах зимней сессии на 1 курсе. Написать программу, которая печатает фамилии студентов, имеющих задолженность хотя бы по одному предмету и "качество" успеваемости.

Входные данные: в каждой строке входных данных в файле INPUT.TXT записаны сведения об одном студенте-первокурснике (всего их не более 30) в формате: <фамилия> <номер группы> <оценка1> <оценка2> <оценка3> причем в фамилии – не более 12 букв, номер группы – целое от 101 до 116, каждая оценка – это 2, 3, 4 или 5, причем первая оценка – за экзамен по матанализу, вторая – по алгебре, третья – по программированию.

Выходные данные: необходимо выдать запись в файл OUTPUT.TXT строки – фамилии студентов, имеющих задолженность хотя бы по одному предмету и целое число – "качество" успеваемости, т.е. процент студентов, сдавших все экзамены на 5 и 4.

Формат вывода:

Failed examinations:

<фамилии>

Progress:

<"качество" успеваемости> %

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Иванов 101 5 4 4 Сидоров 102 3 2 4 Петров 106 4 5 3	Failed examinations: Сидоров Progress: 33 %

7. Вхождение слова в текст

Напишите программу, которая находит первое самое длинное слово и определяет, сколько раз оно встретилось в тексте.

Входные данные: текст, заканчивающийся символом косой черты / в отдельной строке, расположенный в файле INPUT.TXT.

Выходные данные: необходимо выдать в файл OUTPUT.TXT строку – первое самое длинное слово в тексте и целое число – количество вхождений слова в текст.

Формат вывода:

Word:

<первое самое длинное слово>

Number:

<Количество вхождений слова в текст>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Двадцать первое. Ночь. Понедельник. Очертанья столицы во мгле. Сочинил же какой-то бездельник, Что бывает любовь на земле. И от лености или от скуки Все поверили, так и живут: Ждут свиданий, боятся разлуки И любовные песни поют. Но иным открывается тайна, И почиет на них тишина... Я на это наткнулась случайно И с тех пор все как будто больна. /	Word: понедельник Number: 1

8. Форматирование текста

Напишите программу, печатающую символы входного файла в выходной файл так, чтобы строки текста не выходили правее N-й позиции. Это значит, что каждая строка, длина которой превышает N, должна печататься с переносом на следующие строки. Место переноса следует "искать" после последнего символа, отличного от символа-разделителя, расположенного левее N-й позиции. Позаботьтесь о том, чтобы ваша программа вела себя разумно в случае очень длинных строк, а также когда до N-й позиции не встречается ни одного символа пробела или табуляции.

Входные данные: в первой строке файла INPUT.TXT указано целое число – N. Начиная со второй строки, входной файл содержит текст для форматирования.

Выходные данные: необходимо выдать в файл OUTPUT.TXT отформатированный текст.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	30 Я путешествовал без всякой цели, без плана; останавливался везде, где мне нравилось, и отправлялся тотчас далее, как только чувствовал желание видеть новые лица – именно лица.	Я путешествовал без всякой цели, без плана; останавливался везде, где мне нравилось, и отправлялся тотчас далее, как только чувствовал желание видеть новые лица – именно лица.

9. Турне

Текстовый файл содержит сведения о кольцевых туристических маршрутах и их стоимости. Требуется найти маршрут с наибольшим количеством городов и указать его стоимость.

Входные данные: в каждой строке входных данных в файле INPUT.TXT записан маршрут и его стоимость.

Выходные данные: необходимо выдать в файл OUTPUT.TXT строку – маршрут с наибольшим количеством городов и целое число – стоимость турне. Формат вывода:

```
Tour:  
<маршрут>  
Cost:  
<стоимость>
```

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Челябинск – Екатеринбург – Казань – Москва – Самара – Уфа – Челябинск. 5000 Москва – Санкт-Петербург – Москва. 2000	Tour: Москва – Санкт-Петербург – Москва Cost: 2000

10. Перестановка крайних компонент

Дан целочисленный файл, содержащий не менее трех компонент. Поменять в нем первую и последнюю компоненты местами.

Входные данные: в файле INPUT.TXT записаны целые числа (не менее трех).

Выходные данные: необходимо выдать в файл OUTPUT.TXT целые числа – компоненты исходного файла, в котором первая и последняя компоненты переставлены.

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	3 56 -76 0 17 49	49 56 -76 0 17 3

11. Таких не берут в космонавты

Дан файл, содержащий список претендентов в отряд космонавтов. Претендент должен удовлетворять следующим требованиям: рост от 150 до 190 сантиметров, рост в положении сидя не более 90 сантиметров, вес от 50 до 90 килограмм, возраст до 33 лет.

Необходимо выдать в файл сначала список претендентов проходящих формальные требования, а за ним список претендентов, не удовлетворяющих требованиям.

Входные данные: в файле INPUT.TXT записаны данные о претендентах в формате <Фамилия инициалы> <возраст> <рост> <рост сидя>.

Выходные данные: необходимо выдать в файл OUTPUT.TXT записи о претендентах в формате:

Прошли отсев:

<Фамилия инициалы> <возраст> <рост> <рост сидя>

Не прошли отсев:

<Фамилия инициалы> <возраст> <рост> <рост сидя>

Примеры входных и выходных данных

№ п/п	Входные данные	Выходные данные
1	Иванов И.И. 23 156 75 Петров П.П. 28 192 80 Серова Е.О. 30 165 85	Прошли отсев: Иванов И.И. 23 156 75 Серова Е.О. 30 165 85 Не прошли отсев: Петров П.П. 28 192 80

3. ВОПРОСЫ ДЛЯ САМОКОНТРОЛЯ

3.1. Простые задачи

1. Напишите синтаксис выражения.
2. Напишите синтаксис оператора присваивания.
3. Покажите в тексте программы: спецификацию программы, заголовок программы, тело программы, секцию определения переменных программы.
4. Объясните назначение операторов Write, WriteLn, Read, ReadLn. В чем отличие операторов Write и WriteLn? В чем отличие операторов Read и ReadLn?
5. Вычислите (письменно) и объясните полученные результаты следующих выражений:

```
10 div 3
10 div -3
-10 div 3
-10 div -3
10 mod 3
10 mod -3
-10 mod 3
-10 mod -3
```

6. Укажите порядок выполнения операций в выражении (расставьте скобки): $A \text{ mod } B + C \text{ div } D * E$.

3.2. Условный оператор и оператор выбора

1. Напишите синтаксис условного оператора и оператора выбора (с помощью формул Бэкуса-Наура или синтаксических диаграмм).
2. Напишите семантику условного оператора и оператора выбора.
3. Напишите, какое значение будет иметь переменная Z после выполнения следующих операторов

```
Z:=0;
if X>0 then if Y>0 then Z:=1 else Z:=2;
```

при следующих значениях переменных X и Y:

- a) $X = 1, Y = 1$
- b) $X = 1, Y = -1$
- c) $X = -1, Y = 1$

4. Пусть имеется оператор присваивания

```
X:=A or B and C,
```

где переменные X, A, B, C имеют тип Boolean. Напишите условный оператор, который эквивалентен данному (например, оператору присваивания $X := \text{not } A$ эквивалентен условный оператор $\text{if } A \text{ then } X := \text{False} \text{ else } X := \text{True}$).

5. Пусть имеется условный оператор

```
if A then X:=B else X:=C,
```

где переменные X, A, B, C имеют тип Boolean. Напишите оператор присваивания, который эквивалентен данному.

6. Какие из следующих фрагментов программ содержат синтаксические ошибки? Обоснуйте ответ.

- a)

```
var
  S: String;
case S of
  'Zero': WriteLn('Ноль');
  'One': WriteLn('Один');
  'Two': WriteLn('Два');
else
  WriteLn('Много!');
end;
```
- b)

```
var
  B: Boolean;
case B of
  True: WriteLn('Истина');
  False: WriteLn('Ложь');
else
  WriteLn('Неизвестно');
end;
```
- c)

```
var
  R: Real;
case R of
  3.14159: WriteLn('Число Пи');
  2.71828: WriteLn('Число E');
else
  WriteLn('Неизвестная константа');
end;
```
- d)

```
var
  I: Integer;
  S: String;
case I div 100 mod 10 of
  0, 5..9: S:='копеек';
  1: S:='копейка';
  2..4: S:='копейки';
else
  S:='????';
end;
```
- e)

```
type
  DayOfWeek=(Mon, Tue, Wed, Thu, Fri, Sat, Sun);
var
  D: DayOfWeek;
case D of
  Sat..Sun: WriteLn('Выходной день');
else
  WriteLn('Рабочий день');
end;
```

3.3. Операторы цикл

1. Напишите синтаксис операторов следующих операторов цикла: с предусловием, с постусловием, с заранее известным числом шагов.

2. Напишите алгоритмы работы операторов цикла: с предусловием, с постусловием, с заранее известным числом шагов.

3. Определите значение переменной S после выполнения следующих операторов:

- a) S:=0;
i:=0;
while i<5 do
 i:=i+1;
 s:=s+1/i;
- b) S:=0;
i:=1;
while i>1 do begin
 s:=s+1/i;
 i:=i-1;
end;
- c) s:=0;
i:=1;
repeat
 s:=s+1/i;
 i:=i-1;
until i≤1;
- d) s:=1;
n:=1;
for i:=2 to n do
 s:=s+1/i;

4. Напишите четыре варианта функции, возвращающей факториал заданного целого числа, используя следующие операторы цикла

```
for ... to,  
for ... downto,  
while ... do,  
repeat ... until.
```

3.4. Массивы

1. Дайте определение структурного типа массив.
2. Напишите синтаксис определения типа Массив и синтаксис обращения к элементу типа.
3. Для каждого использующего вхождение каждого идентификатора программы укажите его определяющее вхождение.
4. Укажите область видимости и время жизни каждой переменной программы.

3.5. Записи

1. Дайте определение структурного типа Запись.
2. Напишите синтаксис определения типа Запись и синтаксис доступа к элементу типа.
3. Верно ли, что все поля типа данных запись (record) должны быть разных типов? Обоснуйте ответ.
4. Могут ли имена полей типа данных запись (record) совпадать с именами переменных, констант и др. объектов программы? Обоснуйте ответ.
5. Могут ли имена полей одного типа данных запись (record) совпадать с именами полей другого типа данных запись (record)? Обоснуйте ответ.

6. Пусть определены типы данных

```
type
  TDate=record
    Day, Month, Year: Integer;
  end;
  TPerson=record
    Name: String;
    Birth: TDate;
  end;
и переменная
var
  P: TPerson;
```

Используя оператор присоединения with, напишите оператор, в результате выполнения которого переменной P будет присвоено значение

Имя: Иванов

Дата рождения: 16.03.2002 .

3.6. Текстовые файлы

1. Чем отличаются текстовые, типизированные и нетипизированные файлы?
2. Объясните семантику следующих стандартных процедур для работы с текстовыми файлами: Assign, Reset, Rewrite, Append, Close, Eof, Erase, FilePos, FileSize, Rename, Seek, Read, ReadLn, Write, WriteLn.

БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Вирт, Н. Паскаль: Руководство пользователя и описание языка / К. Йенсен, Н. Вирт. – М.: Компьютер, 1995. – 302 с.
2. Ван Тассел, Д. Стил, Разработка, эффективность, отладка и испытание программ / Д. Ван Тассел – М.: Мир, 1985. – 332 с.
3. Лишнер, Р. Delphi: Справочник / Р. Лишнер; пер. с англ. Л. Фрейдина – СПб.: Символ-Плюс, 2001. – 604 с.
4. Себеста, Р. У. Основные концепции языков программирования. / Роберт У. Себеста; пер. с англ. – 5-е изд. – М.: Издательский дом «Вильямс», 2001. – 659 с.
5. Макконнелл, С. Совершенный код. Мастер-класс / С. Макконнелл; пер. с англ. – 2-е изд. – М.: Издательско-торговый дом «Русская редакция» – СПб.: Питер, 2005. – 896 с.
6. Миков, А.И. Информатика. Введение в компьютерные науки / А.И. Миков, Л.Н. Королев. – М.: Высшая школа, 2003. – 342 с.
7. Павловская, Т.А. Паскаль. Программирование на языке высокого уровня: учебник для вузов / Т.А. Павловская – СПб.: Питер, 2003. – 400 с.
8. Пильщиков, В.Н. Сборник упражнений по языку Паскаль: учебное пособие для вузов / В.Н. Пильщиков – М.: Наука, 2002. – 115 с.

ПРИЛОЖЕНИЯ

Приложение 1. Требования и рекомендации по оформлению исходных текстов программ

1.1. Соглашения по идентификаторам

Подбор идентификаторов

1. Все идентификаторы должны выбираться из соображений читаемости и максимальной семантической нагрузки. Например:

- 1) const
Eps = 0.0001; { Точность }
- 2) var
Sum: Integer; { Сумма }
Message: String; { Сообщение }

Неудачными можно считать идентификаторы:

- 1) const
E = 0.0001; { Точность }
- 2) var
K: Integer; { Сумма }
S: String; { Сообщение }

2. Идентификаторы рекомендуется подбирать из слов английского языка.

Например:

- 1) procedure Beep(Hertz, MSec: Word);
{Выдает звуковой сигнал заданной частоты и длительности }
- 2) function ExistFile(FName: String): Boolean;
{ Выдает True, если файл с именем FName существует }
- 3) var
Done: Boolean; { Признак окончания работы с программой }
Width, Height: Real; { Размеры изделия: ширина, высота }

Не очень удачными можно считать идентификаторы:

- 1) procedure Zvuk(Chast, Dlit: Word);
{Выдает звуковой сигнал заданной частоты и длительности }
- 2) function EstFile(Im: String): Boolean;
{ Выдает True, если файл с именем Im существует }
- 3) var
Konec: Boolean; { Признак окончания работы с программой }
Shirina, Vysota: Real; {Размеры изделия: ширина, высота }

Написание идентификаторов

1. Зарезервированные слова языка Pascal следует писать только маленькими буквами. Например:

type, var, const; procedure, function; begin, end; if, then, else; repeat, until; case; while; and, not, or; unit, uses, interface, implementation.

2. В любых идентификаторах каждое слово, входящее в идентификатор, следует писать, начиная с большой буквы, остальные буквы – маленькие. В этом смысле символ подчеркивания "_" между словами идентификатора можно считать излишним. Например:

```

1) var
    NextX, LastX: Real; { следующая и предыдущая итерация }
    BeepOnError: Boolean; { подавать ли звуковой сигнал при
    неправильном вводе пользователя? }
2) function GraphErrorMsg(ErrCode: Word): String;
    { стандартная функция модуля Graph; выдает описание ошибки
    использования графики по ее коду }
3) begin
    Write('Введите целое число, большее 10 ==>_ ');
    ReadLn(Num);
    if Num > 10 then WriteLn('OK');
    end.
4) var
    FileName: String; { а не File_Name }
    type
    ByteArray = array [0..65534] of Byte; { а не Ar-
    ray_Of_Byte }

```

3. Рекомендуется идентификаторы, состоящие из одной буквы, писать большой буквой, если они не являются счетчиками циклов. Например:

```

1) A := A + B;
   B := A - B;
   A := A - B;
2) procedure DeleteNode(var L: List; I: Word);
    { удалить узел с номером I из списка L }
3) for i := 1 to N do
    for j := 1 to N do
        A[i, j] := (i + j) / 2;

```

1.2. Соглашения по самодокументируемости программ

Комментарии

1. Комментарии в теле программы следует писать на русском языке так, чтобы программист, не участвовавший в разработке программы (но имеющий опыт работы на языке Pascal), мог без особого труда разобраться в логике программы, и при необходимости, сопровождать данный программный продукт.

2. Рекомендуется комментарии программы писать внутри символов { и }, а символы (* и *) использовать при отладке программы как "заглушки" участков программного кода.

Спецификация пользовательской процедуры или функции

Для каждой пользовательской процедуры или функции должна быть описана в виде комментария спецификация, содержащая следующую информацию:

- назначение процедуры или функции;
- описание семантики параметров-значений (параметров, передаваемых по значению), если она неочевидна;
- описание семантики параметров-переменных (параметров, передаваемых по ссылке), если она неочевидна.

- для функции: описание семантики возвращаемого значения, если она неочевидна.

Например:

1) Семантика параметров и возвращаемого значения очевидна

```
function IsLeapYear(Y: Word): Boolean;  
{ возвращает True, если год Y -- високосный }
```

2) Семантика параметров очевидна, семантика возвращаемого значения неочевидна

```
function DayOfWeek(D, M, Y: Word): Byte;  
{ Возвращает день недели даты D/M/Y;  
год Y должен быть в отрезке 1582..4902;  
результат: ВСК = 0, Пнд = 1, Втр = 2, ... Сбт = 6 }
```

3) Семантика параметров и возвращаемого значения неочевидна

```
const  
  MaxN = 10; type  
  Matrix = array [1..MaxN, 1..MaxN] of Real;  
  Vector = array [1..MaxN] of Real;  
procedure Gauss(A: Matrix; B: Vector; Eps: Real;  
  var X: Vector;  
  var HasSolution: Boolean;  
  var NumOfRoots: Integer;  
  var Det: Real;  
  var AForReverse: Matrix; var BForReverse: Vector);  
{ Решение системы линейных алгебраических уравнений методом Га-  
усса.
```

Входные данные:

A -- матрица коэффициентов системы;
B -- столбец свободных членов системы;
Eps -- точность вычислений.

Выходные данные:

X вектор решения;
HasSolution флаг, устанавливаемый в TRUE, если решение системы существует, и в FALSE в противном случае;
NumOfRoots число корней в решении системы, может принимать значения:
0 если решение системы не существует,
MaxN если решение системы существует и единственно,
MaxInt если существует бесконечное множество решений;
Det -- значение определителя матрицы A;
AForReverse -- нижняя треугольная матрица, полученная из A в результате выполнения прямого хода алгоритма Гаусса;
BForReverse -- столбец свободных членов, полученный из B в результате выполнения прямого хода алгоритма Гаусса. }

Замечание: Если процедура (функция) реализует какой-либо вычислительный метод (например: нахождение площади фигуры методом трапеций, поиск минимума функции методом Ньютона и т.п.), необходимо в теле процедуры (функции) поместить комментарий с кратким описанием метода, либо ссылку на источник, где описан метод.

Спецификация пользовательского программного файла или модуля

Программный файл или модуль (unit) должен начинаться со спецификации в виде комментария, содержащего следующую информацию:

- имя файла;
- фамилия автора;
- дата написания файла;
- версия языка и замечания по компиляции программы (модуля) в других версиях языка;
- назначение программы (модуля);

Например:

1) { PRIMES.PAS

```
-----  
(с)оздал   : Иванов И.И.  
дата      : 01/09/09  
для       : Turbo Pascal 6.0  
-----
```

Подсчет количества простых чисел в промежутке [1..200]. }

2) { MYMENU.PAS

```
-----  
(с)оздал   : Иванов И.И.  
дата      : 01/09/09  
для       : Turbo Pascal 6.0  
-----
```

Модуль экспортирует процедуры и функции, поддерживающие горизонтальные и вертикальные меню. }

Замечание: В программном файле после заголовка "Program <имя программы>," (в файле с пользовательским модулем — после заголовка "unit <имя модуля>,") рекомендуется поместить комментарий с указаниями по запуску программы и работе с ней (указаниями по использованию модуля другими программистами) или ссылку на источник, который использован при составлении программы (модуля).

1.3. Соглашения по читаемости программ

1. Для отражения структурной вложенности языковых конструкций в тексте программы должна использоваться "лесенка" (отступы от левого края). Рекомендуется отступ лесенки не менее 2-х и не более 4-х пробелов.

Принятого отступа нужно придерживаться во всем тексте программы. Правильное написание операторов языка Pascal приведено в таблице 1.

Таблица 1

Правильное написание операторов языка Pascal

Оператор	Правильное написание
if	if условие then оператор;
	if условие then begin операторы; end;
	if условие then begin операторы; end;

Оператор	Правильное написание
	<pre> if условие then begin операторы; end else begin операторы; end; </pre>
while	<pre> while условие do оператор; </pre>
	<pre> while условие do begin операторы; end; </pre>
	<pre> while условие do begin операторы; end; </pre>
for	<pre> for счетчик := нач to кон do оператор; </pre>
	<pre> for счетчик := нач to кон do begin операторы; end; </pre>
	<pre> for счетчик := нач to кон do begin операторы; end; </pre>
case	<pre> case выражение of выражение: оператор; end; </pre>
	<pre> case выражение of выражение: оператор; else оператор; end; </pre>
	<pre> case выражение of выражение: begin операторы; end; end; </pre>
	<pre> case выражение of выражение: begin операторы; end; end; </pre>

Например:

```
1) function Sign(X: Real): Integer;
{ выдает знак числа X }
begin
  if X > 0 then
    Sign := 1
  else
    if X < 0 then
      Sign := -1
    else
      Sign := 0;
end;

2) procedure Equation(A, B, C: Real; var X1, X2: Real;
var Num: Byte);
{ нахождение действительных корней квадратного уравнения;
A, B, C -- коэффициенты
X1, X2 -- корни (если действительного решения нет, то полагаются
равными 0);
Num -- число корней (0, 1, или 2) }
var
  D: Real;
begin
  D := Sqr(B) - 4 * A * C;
  if D < 0 then begin
    Num := 0;
    X1 := 0;
    X2 := 0;
  end
  else begin
    X1 := (- B + Sqrt(D)) / (2 * A);
    X2 := (- B - Sqrt(D)) / (2 * A);
    if X1 = X2 then Num := 1
    else
      Num := 2;
  end;
end;
```

2. Длина строк программы не должна превышать ширины экрана (80 символов).

3. Рекомендуется операнды бинарных операций отделять от знака операции одним пробелом. Например: Sum := A + B;

4. Рекомендуется при перечислении идентификаторов после запятой ставить один пробел. Например:

```
1) WriteLn('Сумма: ', A + B, ';', 'Разность: ', A - B, '.');
2) var
  Day, Month, Year: Word;
  i, j, k, l, m, n: Integer;
```

5. Рекомендуется всегда писать символ-разделитель операторов ";" непосредственно после оператора. Например:

```
1) case Num of
  1: WriteLn('один...');
  2: WriteLn('два...');
  3: WriteLn('три...'); { <-- здесь }
else
  WriteLn('много!'); { <-- и здесь }
end;
2) if N < 0 then
begin
  WriteLn('Введено неверное значение N, прерываем работу!');
  Halt; { <-- здесь }
end;
```

Приложение 2. Пример оформления практической работы

Иванов Иван
Группа ВМИ-117

Домашняя работа № 2 – Определение вида треугольника по трем сторонам

1. Условие задачи

Заданы длины сторон треугольника. Если треугольник с такими сторонами существует, определить его вид: равносторонний, равнобедренный, общий. Иначе сообщить, что треугольник не существует.

Входные данные

Три вещественных числа (не обязательно положительные!) – длины сторон треугольника.

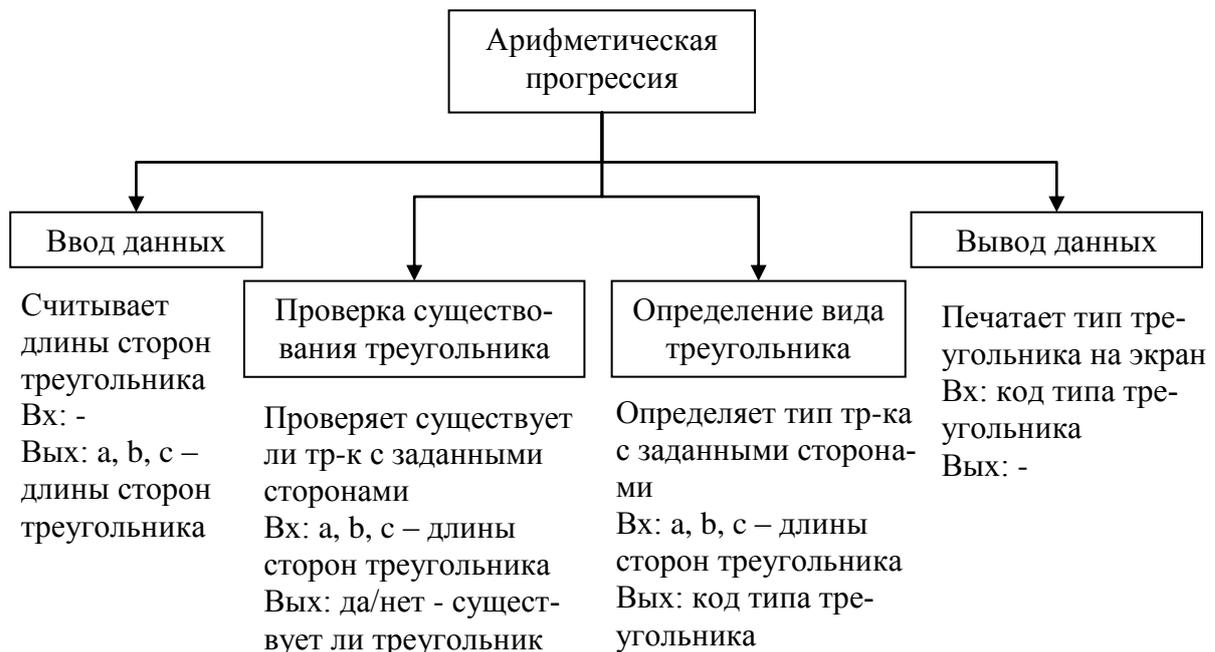
Выходные данные

Одно из сообщений: РАВНОСТОРОННИЙ, РАВНОБЕДРЕННЫЙ, ОБЩЕГО ВИДА, НЕ СУЩЕСТВУЕТ.

Пример тестовых данных:

№	Входные данные	Выходные данные
1	1 1 1	РАВНОСТОРОННИЙ
2	1 2 3	НЕ СУЩЕСТВУЕТ
3	3 4 5	ОБЩЕГО ВИДА

2. Модульная структура



3. Листинг

```
{ TRIANGLE.PAS  
Нахождение вида треугольника.  
(с) Иванов И.И. (ММ-156)  
11-дек-13 }
```

```
program Triangle;
```

```

{ Запрос у пользователя и ввод длин трех сторон треугольника с клавиатуры.}
procedure readTriangleSides(var a, b, c: real);
begin
  Write('Введите длины сторон треугольника: ');
  ReadLn(a, b, c);
end;

{ Определение существования треугольника по заданным длинам сторон.
Если треугольник существует, возвращает TRUE, иначе FALSE. }
function isTriangleExist(a, b, c: real): boolean;
begin
  if (a + b > c) and (a + c > b) and (b + c > a) then
    Result := True
  else
    Result := False;
end;

{ Определение вида треугольника по заданным длинам сторон.
Возвращает:
3 треугольник равносторонний
2 треугольник равнобедренный
1 треугольник общего вида
Треугольник с заданными сторонами заведомо должен существовать. }
function getTriangleKind(a, b, c: real): integer;
begin
  if (a = b) and (b = c) then
    Result := 3
  else if (a = b) or (a = c) or (b = c) then
    Result := 2
  else
    Result := 1;
end;

{ Вывод на экран сообщения о виде треугольника в зависимости от значения K:
0 ТРЕУГОЛЬНИК НЕ СУЩЕСТВУЕТ
1 ТРЕУГОЛЬНИК ОБЩЕГО ВИДА
2 ТРЕУГОЛЬНИК РАВНОБЕДРЕННЫЙ
3 ТРЕУГОЛЬНИК РАВНОСТОРОННИЙ }
procedure printTriangleType(triangleKind: integer);
begin
  case triangleKind of
    0: WriteLn('ТРЕУГОЛЬНИК НЕ СУЩЕСТВУЕТ');
    1: WriteLn('ТРЕУГОЛЬНИК ОБЩЕГО ВИДА');
    2: WriteLn('ТРЕУГОЛЬНИК РАВНОБЕДРЕННЫЙ');
    3: WriteLn('ТРЕУГОЛЬНИК РАВНОСТОРОННИЙ');
  end;
end;

end;

var
  {длины сторон треугольника}
  a, b, c: real;
  triangleExist: boolean;
  triangleKind: integer;

```

```

begin
  readTriangleSides(a, b, c);
  triangleExist := isTriangleExist(a, b, c);
  if triangleExist then
    triangleKind := getTriangleKind(a, b, c)
  else
    triangleKind := 0;
  printTriangleType(triangleKind);
  Readln;
end.

```

4. Тесты

№	Входные дан- ные	Ожидаемый результат	Действительный ре- зультат	Тест пройден
1	1 1 1	РАВНОСТОРОННИЙ	РАВНОСТОРОННИЙ	Да
2	1 2 3	НЕ СУЩЕСТВУЕТ	ОБЩЕГО ВИДА	Нет
3	3 4 5	ОБЩЕГО ВИДА	ОБЩЕГО ВИДА	Да
4	0 0 0	НЕ СУЩЕСТВУЕТ	РАВНОСТОРОННИЙ	Нет
5	4 4 5	РАВНОБЕДРЕННЫЙ	РАВНОБЕДРЕННЫЙ	Да
6	2 1 1	НЕ СУЩЕСТВУЕТ	РАВНОБЕДРЕННЫЙ	Нет
7	-1 -1 -1	НЕ СУЩЕСТВУЕТ	Runtime error 106 at 0ВЕВ:0026.	Нет

ОГЛАВЛЕНИЕ

Введение.....	3
1. Техника разработки программ	4
2. Задания и методические указания по их выполнению	
2.1. Простые задачи	6
2.2. Условный оператор и оператор выбора.....	11
2.3. Операторы цикла.....	18
2.4. Массивы.....	23
2.5. Записи.....	33
2.6. Текстовые файлы.....	44
3. Вопросы для самоконтроля	
3.1. Простые задачи	51
3.2. Условный оператор и оператор выбора.....	51
3.3. Операторы цикл.....	52
3.4. Массивы.....	53
3.5. Записи.....	53
3.6. Текстовые файлы.....	54
Библиографический список	55
Приложения	56
Приложение 1. Требования и рекомендации по оформлению исходных текстов программ	56
Приложение 2. Пример оформления практической работы	63