Федеральное агентство по образованию
Удмуртский государственный университет

А.Е. Анисимов

СБОРНИК ЗАДАЧ ПО НАЧАЛАМ ПРОГРАММИРОВАНИЯ

Ижевск 2004

УДК 681.3.06(075)
ББК 32.973-018я7
А67

Рецензенты: доктор физико-математических наук, профессор кафедры алгебры и топологии УдГУ **Н.Н. Непейвода**;
кандидат физико-математических наук, доцент, заведующий кафедрой высшей математики УдГУ **В.И. Родионов**

Анисимов А.Е.
А67 Сборник задач по началам программирования/УдГУ. Ижевск, 2004. 149 с.

Сборник задач предназначен для студентов младших курсов вузов, учащихся школ, изучающих основы программирования. Задачи могут быть использованы преподавателями, учителями в таких учебных курсах, как «Информатика», «Компьютерный вычислительный практикум», «Основы программирования», «Компьютерные науки» и других. Кроме этого, он может быть полезен всем, кто решил самостоятельно освоить основы программирования на ЭВМ.

УДК 681.3.06(075)
ББК 32.973-018я7

Предисловие

В предлагаемом сборнике задач по началам программирования содержится более 650 задач по различным темам. Материал задач сборника может быть использован студентами и преподавателями, старшеклассниками и учителями в учебном процессе таких дисциплин, как «Информатика», «Компьютерный вычислительный практикум», «Основы программирования», «Компьютерные науки» и других; кроме этого, он может быть полезен всем, кто решил самостоятельно освоить основы программирования на ЭВМ.

Материал сгруппирован по разделам. Каждый раздел посвящен использованию одной из конструкций языка программирования, определенной технологии или предметной области. В разделе не менее 25 задач, что позволит преподавателю, учителю распределить их между обучающимися в группе в виде лабораторных компьютерных заданий. Внутри каждого раздела задачи по сложности разбиты на три группы – A, B и C. Группа A содержит, в основном, стандартные задачи данной тематики. Группа B состоит из задач повышенной (по отношению к группе A) сложности. Группа C также содержит задачи повышенной сложности, при решении которых возможно использование каких-либо нестандартных приемов и методов. Некоторые задачи группы C предлагались на различных олимпиадах по программированию.

Ряд задач разделены на пункты а), б) и т.д. В некоторых случаях каждый пункт представляет собой отдельную задачу, в других – это элемент единого задания. В любом случае решение о распределении задач принимает преподаватель (учитель). В конце сборника для нескольких задач приведены тесты, решения или подсказки.

Постановки задач, как правило, не ориентированы на конкретный язык программирования. В качестве инструмента могут быть привлечены различные системы: Turbo Pascal, Think

Pascal, C/C++, Basic, Visual Basic, Delphi и другие. Следует обратить внимание, что цель данного сборника – не изучение деталей использования конкретной языковой системы, а освоение методов и приемов программирования «вообще». Изучение же конкретных языковых сред (в том числе современных, объектно-ориентированных) будет более эффективно при наличии багажа знаний, умений и навыков, полученных при решении задач по основам программирования.

Большинство задач не требует привлечения каких-либо специальных знаний из области математики, в основном достаточно школьного уровня и/или материала первого курса вуза. Хотя в тексте некоторых задач явно приведены формулы, никогда и никому не помешает лишний раз обратиться к дополнительному справочному материалу.

Многие задачи являются классическими в каждой конкретной теме. Идеи некоторых задач отчасти заимствованы из других сборников, правда, формулировки, условия несколько изменены. Список рекомендуемой литературы приведен в конце сборника.

Автор-составитель считает своим долгом выразить благодарность С. А. Канторовичу и В.В. Пупышеву за оказанную помощь, полезные советы и идеи, которые весьма пригодились при создании сборника.

Автор будет благодарен всем, кто направит свои замечания, пожелания, предложения и вопросы, касающиеся текста и содержания задачника по адресу электронной почты *aae@ulm.uni.udm.ru*.

Задачи

1. Выражения. Оператор присваивания

Для решения большинства задач этого раздела необходимо написать программу, использующую, помимо ввода и вывода, *только* операторы присваивания.

А

1.1. Расставьте скобки в выражении, приведенном справа, так, чтобы последовательность выполняемых операций соответствовала записи математического выражения (слева):

а) $\dfrac{1}{1+\dfrac{1}{2+\dfrac{1}{3+\dfrac{1}{4}}}}$ $\quad 1/1+1/2+1/3+1/4$;

б) $\dfrac{a+b}{x-y}c-y$ $\quad a+b*c/x-y-y.$

1.2. Сформулировать на языке программирования логическое выражение, истинное при выполнении следующего условия, и ложное – в противном случае:

а) $a = max(x, y, z)$;

б) целые m и n имеют одинаковую четность;

в) целые p и q могут быть соответственно числителем и знаменателем правильной несократимой обыкновенной дроби;

г) уравнение $ax^2 + bx + c = 0$ имеет в точности один вещественный корень;

д) система из двух линейных уравнений имеет в точности одно решение;

е) переменные a и b имеют одинаковые логические значения;

ж) в точности две из трех переменных a, b и c имеют одинаковые логические значения;

з) из клетки (a_1, b_1) шахматной доски можно попасть в клетку (a_2, b_2) за один ход

- ферзя;
- коня.

1.3. Обмен значениями. Даны значения двух переменных a и b. Поменять местами значения этих переменных.

1.4. Тройной обмен. Даны значения трех переменных a, b и c. Поменять местами значения этих переменных так, чтобы a получила бы значение b, b – значение c, а c – исходное значение a.

1.5. Дано положительное число r, натуральное n $(n > 2)$. Найти площадь правильного n-угольника, вписанного в круг радиуса r.

1.6. Даны координаты трех вершин треугольника $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$. Найти площадь треугольника ABC.

1.7. Даны координаты трех вершин треугольника $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$. Найти координаты точки пересечения биссектрис.

1.8. Даны координаты трех точек $A(x_1, y_1)$, $B(x_2, y_2)$ и $C(x_3, y_3)$. Найти координаты точки D , если $\overrightarrow{AB} + \overrightarrow{AC} = \overrightarrow{AD}$.

1.9. Даны длины сторон треугольника a, b и c. Найти площади вписанной и описанной окружностей этого треугольника.

1.10. Даны вещественные числа a, b и c. Найти координаты вершины параболы $y = ax^2 + bx + c$.

1.11. Даны a, b, α. Найти площадь треугольника со сторонами a и b и углом α между ними, выраженным в градусах.

1.12. Дан объём куба V. Найти площадь поверхности и длину ребра куба, объём вписанного в куб шара.

1.13. Дано число . Найти:
- а) a^4 за две операции умножения;
- б) a^6 за три операции умножения;
- в) a^5 за три операции умножения;
- г) a^{10} за четыре операции умножения;
- д) a^{21} за шесть операций умножения;
- е) a^{28} за шесть операций умножения;
- ж) a^{32} за пять операций умножения;
- з) a^{40} за шесть операций умножения;
- и) a^{48} за шесть операций умножения;
- к) a^{36} за шесть операций умножения;
- л) a^{64} за шесть операций умножения;
- м) a^{104} за восемь операций умножения.

1.14. Дано вещественное x. Найти значение выражения $16x^4 - 6x^3 + 4x^2 - x + 1$. Разрешается использовать не более 5 операций умножения.

1.15. Дано вещественное x. Найти значение выражения $x^4 - 2x^3 + x^2 - 2x + 1$. Разрешается использовать не более 4 операций умножения.

1.16. Дано вещественное x. Найти значение выражения $8x^7 + 4x^4 + 2x^3 - 2x^2 + x - 1$. Разрешается использовать не более 5 операций умножения.

1.17. Дано время N в секундах. Выразить это время в сутках, часах, минутах и секундах.

1.18. Даны целые неотрицательные h_1, m_1, h_2, m_2 ($h_1 \in [0, 23]$; $m_1, m_2 \in [0, 59]$) . Поезд вышел первого числа со станции отправления в h_1 часов m_1 минут и был в пути до станции назначения h_2 часов m_2 минут. Определить число и время прибытия поезда на станцию назначения.

1.19. Даны натуральные n и a. Указать номер дня недели n-го числа месяца, у которого первое число имеет номер дня недели a (понедельник – первый день недели, воскресенье – седьмой).

1.20. Перевести длину f футов i дюймов в целые метры, сантиметры и миллиметры (1 фут = 12 дюймов, 1 дюйм = 25,4 миллиметра).

1.21. Даны целые неотрицательные h, m и s. Найти углы поворота в градусах часовой, минутной и секундной стрелок стрелочных часов, когда они показывают время h часов, m минут и s секунд.

1.22. Дан угол, выраженный в радианах. Выразить угол в целых градусах, минутах и секундах с максимальной точностью.

1.23. Даны вещественные a и b. Выразить комплексное число $z = a + bi$ в тригонометрической форме $z = r(\cos\varphi + i\sin\varphi)$.

1.24. Дано целое трехзначное число a. Найти число b, полученное из a перестановкой цифр в обратном порядке.

1.25. Дано целое трехзначное число a. Найти сумму цифр этого числа.

1.26. Свечки для торта. Родители купили ребенку набор свечей для торта, количество свечей в наборе – N (натуральное число). Когда ребенку исполнился один год, на торте была одна свечка, два года – две, и т.д. На сколько лет хватит приобретенного набора свечей?

B

1.27. Даны координаты четырех точек $A(x_1, y_1)$, $B(x_2, y_2)$, $C(x_3, y_3)$ и $D(x_4, y_4)$. Присвоить логической переменной b значение «ИСТИНА», если точка D лежит внутри треугольника ABC, и «ЛОЖЬ» – в противном случае.

1.28. Даны координаты двух точек $A(x_1, y_1)$ и $B(x_2, y_2)$, угол α, длина d. Найти координаты точек C и D, если $ABCD$ – параллелограмм, $\angle A = \alpha$, $AD = d$.

1.29. Високосный год. Дано целое неотрицательное N. Присвоить логической переменной V значение «ИСТИНА», если N – високосный год, и «ЛОЖЬ» – в противном случае. Високосным считается год, если он делится на 4, за исключением тех, которые делятся на 100, но не делятся на 400.

C

1.30. Множество натуральных чисел $\{1, 2, 3, \ldots\}$ разобьем на последовательные упорядоченные наборы: в первом наборе один элемент $\{1\}$, во втором – два $\{2, 3\}$, в третьем – три $\{4, 5, 6\}$ и т.д. Дано натуральное N, принадлежащее одному из наборов. Найти номер этого набора и номер, под каким элемент входит в этот набор.

1.31. Множество натуральных чисел разбито на наборы, как в задаче 1.30. Даны два натуральных числа $k, l (l \leq k)$. Какое значение имеет l-й элемент набора с номером k?

1.32. Обмен с ограничением. Решить задачу 1.3 для случая двух целых переменных. Ограничение: не использовать дополнительных переменных.

1.33. Обмен логических. Решить задачу 1.3 для случая двух логических переменных. Ограничение: не использовать дополнительных переменных.

1.34. Обмен множеств. Решить задачу 1.3 для случая двух переменных типа «множество» (см. разд. 8.). Ограничение: не использовать дополнительных переменных.

1.35. Тройной ограниченный обмен. Решить задачу 1.4 для случая трех целых переменных. Ограничение: не использовать дополнительных переменных.

1.36. Никаких «если»! Даны вещественные a и b. Найти наибольшее значение из a и b. Ограничение: разрешается использовать *только* оператор присваивания.

1.37. Снова никаких «если»! Даны вещественные a и b. Найти наименьшее значение из a и b. Ограничение то же, что и в задаче 1.36.

1.38. Даны вещественные a, b и c. Найти наибольшее значение из a, b и c. Ограничение то же, что и в задаче 1.36.

1.39. Даны вещественные a, b и c. Найти наименьшее значение из a, b и c. Ограничение то же, что и в задаче 1.36.

1.40. Даны вещественные a и b. Найти площадь наибольшего круга, который можно вписать в прямоугольник со сторонами a и b. Ограничение то же, что и в задаче 1.36.

1.41. Даны различные вещественные a, b и c. Найти то значение из них, которое лежит на числовой оси между двумя другими. Ограничение то же, что и в задаче 1.36.

1.42. Дано целое трехзначное число a. Найти наименьшее число b, полученное из a перестановкой цифр. Ограничение то же, что и в задаче 1.36.

1.43. Дано целое трехзначное число a. Найти наименьшую и наибольшую цифры a. Ограничение то же, что и в задаче 1.36.

1.44. Даны целые x_1,y_1,x_2,y_2 ($x_1, y_1, x_2, y_2 \in [1; 8]$). Поля шахматной доски задаются координатами (a, b). Присвоить логической переменной b значение «ИСТИНА», если ферзь, находящийся на поле (x_1, y_1), бьет фигуру, находящуюся на поле (x_2, y_2), и «ЛОЖЬ» – в противном случае. Ограничение то же, что и в задаче 1.36.

1.45. Даны целые x_1,y_1,x_2,y_2 $(x_1, y_1, x_2, y_2 \in [1; 8])$. На поле шахматной доски (x_1, y_1) находится слон, на поле (x_2, y_2) – конь. Присвоить логической переменной b значение «ИСТИНА», если одна фигура бьет другую, и «ЛОЖЬ» – в противном случае. Ограничение то же, что и в задаче 1.36.

1.46. Решить задачу 1.45 для ладьи и слона.

1.47. Решить задачу 1.45 для пешки и коня. Считать, что пешка продвигается по доске в направлении увеличения второй координаты. Подсказка: а что происходит с пешкой, если она имеет координаты $(x_1, 8)$?

2. Разветвление

А

2.1. Даны значения вещественных переменных a и b. Перераспределить так значения, чтобы в a оказалось наибольшее, а в b – наименьшее значение.

2.2. Даны значения вещественных переменных a, b и c. Найти среди них наибольшее и наименьшее значения.

2.3. Даны значения трех переменных a, b и c, отличные друг от друга. Обменять значения переменных с наибольшим и с наименьшим значениями.

2.4. Подарок. Даны целые неотрицательные a, b, c, X, Y, Z. Известно, что в каждый новогодний подарок необходимо положить a конфет, b яблок и c груш. Какое максимальное количество подарков можно скомплектовать из X конфет, Y яблок и Z груш.

2.5. Оси. Даны координаты двух точек плоскости $A(x_1, y_1)$, $B(x_2, y_2)$. Определить названия координатных осей, пересекаемых отрезком AB.

2.6. Четверти. Даны координаты двух точек плоскости $A(x_1, y_1)$, $B(x_2, y_2)$. Определить номера четвертей координатной плоскости, которым принадлежат точки отрезка AB.

2.7. Линейное уравнение? Даны вещественные значения b и c. Решить уравнение $bx + c = 0$.

2.8. Квадратное уравнение. Даны значения a, b и c ($a \neq 0$). Решить квадратное уравнение $ax^2 + bx + c = 0$.

2.9. Система линейных уравнений. Даны вещественные a_1, a_2, b_1, b_2, c_1, c_2. Решить систему уравнений

$$\begin{cases} a_1x + b_1y &= c_1 \\ a_2x + b_2y &= c_2. \end{cases}$$

2.10. Даны координаты точки плоскости $A(x, y), xy \neq 0$. Определить номер четверти, которой принадлежит эта точка.

2.11. Даны значения вещественных переменных a и b. Найти, в какой координатной четверти расположен треугольник, образованный прямой, заданной уравнением y=ax+b, и осями координат, если такой треугольник существует.

2.12. Даны числа $x_1, y_1, x_2, y_2, x_3, y_3$. Определить, лежит ли точка (x_1, y_1) внутри или вне прямоугольника, стороны которого параллельны осям координат, а противолежащие вершины имеют координаты (x_2, y_2) и (x_3, y_3).

2.13. Даны координаты точки (x, y), радиус r и координаты центра окружности (x_c, y_c). Определить, лежит ли точка внутри или вне круга, ограниченного указанной окружностью, или на окружности.

2.14. Даны числа $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$. Определить, лежит ли точка (x_1, y_1) внутри или вне треугольника с вершинами $(x_2, y_2), (x_3, y_3), (x_4, y_4)$.

В

2.15. Даны значения вещественных переменных a, b и c. Вывести значения в порядке неубывания.

2.16. Квадратное уравнение? Даны значения a, b и c. Решить уравнение $ax^2 + bx + c = 0$.

2.17. Даны координаты точки плоскости (x, y). Определить область, которой принадлежит эта точка: номер четверти или имя оси или начало координат.

2.18. Даны целые неотрицательные a, X, Y, Z. Известно, что в каждый новогодний подарок необходимо положить a конфет, одно или два яблока, одну или две груши, причем количество фруктов в подарке должно равняться трем. Какое максимальное количество подарков можно скомплектовать из X конфет, Y яблок и Z груш.

2.19. Даны числа a, b и c. Если треугольник с такими сторонами не существует, выдать на экран 0, иначе – выдать 3, 2 или 1 в зависимости от того, равносторонний это треугольник, равнобедренный или какой-нибудь еще.

2.20. Даны вещественные положительные a, b, c, d. Определить, можно ли прямоугольник со сторонами a и b поместить внутри прямоугольника со сторонами c и d так, чтобы каждая сторона одного прямоугольника была параллельна или перпендикулярна каждой стороне другого прямоугольника.

2.21. Сумма прописью. Дано вещественное число $r, (0 \leqslant r < 100)$ с не более чем двумя значащами цифрами после десятичной точки. Считаем, что r обозначает денежную сумму в рублях. Вывести на экран правильно согласованную фразу, обозначающую r, в виде "X рублей Y копеек"(например, число

22.21 должно быть выведено в виде "22 рубля 21 копейка").

2.22. Даны вещественные положительные a, b, c, d, e, f. Считаем, что пары чисел a и b, c и d, e и f обозначают размеры первого, второго и третьего прямоугольника соответственно. Выяснить, можно ли внутри одного из прямоугольников уместить два других, не накладывая один на другой.

2.23. Задача жестянщика. Даны вещественные положительные r, a, b, c и d. Определить, можно ли из круглой заготовки радиусом r вырезать две пластины прямоугольной формы с размерами $a \times b$ и $c \times d$.

2.24. Полпути. Студент добирался домой в другой город: в начале пути на автобусе t_1 часов со скоростью v_1, затем на велосипеде t_2 часов со скоростью v_2, затем пешком t_3 часов со скоростью v_3. На половине пути он делал остановку в придорожном кафе. На чем он тогда передвигался?

C

2.25. Что это? Даны координаты четырех различных точек плоскости, причем никакие три из них не лежат на одной прямой. Считая эти точки вершинами четырехугольника, определить его тип: прямоугольник, квадрат, ромб, трапеция, параллелограмм или произвольный четырехугольник.

2.26. Кирпич. Даны положительные вещественные числа x, y, a, b, c. Определить, пройдёт ли кирпич с рёбрами a, b и c в прямоугольное отверстие в стене со сторонами x и y.

2.27. Даны вещественные положительные a, b, c, d, e, f. Считаем, что пары чисел a и b, c и d, e и f - размеры первого, второго и третьего прямоугольника соответственно. Выяснить, можно ли внутри одного из прямоугольников уместить два других, не накладывая один на другой.

2.28. Биквадратное уравнение. Даны значения a, b и c ($a \neq$

0). Решить биквадратное уравнение $ax^4 + bx^2 + c = 0$.

2.29. Биквадратное уравнение? Даны значения a, b и c. Решить уравнение $ax^4 + bx^2 + c = 0$.

2.30. Дано значение a. Решить уравнение $\cos^2 x - (a-2)\cos x + 4a + 1 = 0$.

2.31. Дано значение a. Решить уравнение $a^2 \sin^2 x + (2a - 1)\sin x + 1 = 0$.

2.32. Дано значение a. Решить уравнение $|x^2 - 8|x| + 7| = a$.

2.33. Дано значение a. Решить уравнение $|x^2 - 6|x| + a| = a$.

2.34. Даны вещественные $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$. Определить, имеет ли отрезок с концами (x_1, y_1) и (x_2, y_2) общие точки с отрезком с концами (x_3, y_3) и (x_4, y_4).

2.35. Даны вещественные $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$. Найти длину части отрезка с концами в точках (x_1, y_1) и (x_2, y_2), находящуюся внутри прямоугольника со сторонами, параллельными осям координат, несоседние вершины которого имеют координаты (x_3, y_3) и (x_4, y_4).

2.36. Даны вещественные числа a, b и c. Найти точные вещественные корни уравнения $ax^2+bx+c = 0$ (то есть если результат выполнения некоторой операции нельзя записать в виде десятичной непериодической дроби, то эту операцию выполнять не нужно).

Пример.

a=*1* b=*-1* c=*-1*
x1=0.5– sqrt(5)/2
x2=0.5+sqrt(5)/2

2.37. Даны вещественные $x_1, y_1, r, x_2, y_2, x_3, y_3$. Определить, существуют ли общие точки у круга с центром в точке (x_1, y_1) и радиусом r и области, ограниченной прямоугольником, стороны которого параллельны или перпендикулярны осям координат, а (x_2, y_2) и (x_3, y_3) – координаты его противоположных вершин.

2.38. Даны четыре точки на числовой прямой A, B, C и D. Найти длину пересечения отрезков AB и CD.

2.39. Даны вещественные $x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4$. Определить, имеет ли отрезок с концами (x_1, y_1) и (x_2, y_2) общие точки с отрезком с концами (x_3, y_3) и (x_4, y_4).

2.40. Даны вещественные $x_1, y_1, x_2, y_2, x_3, y_3$. Найти количество точек треугольника с вершинами (x_1, y_1), (x_2, y_2), (x_3, y_3), принадлежащими осям координат (возможен ответ — «бесконечное множество»).

2.41. Дешевая смесь. Пусть v_1, v_2 — объемы двух растворов, k_1, k_2 — их концентрации, p_1, p_2 — их цены за единицу объема. Пусть требуется получить смесь объемом v_3 с концентрацией k_3. Найти такие объемы смешиваемых растворов v_1' и v_2' ($v_3 = v_1' + v_2', v_1' \leqslant v_1, v_2' \leqslant v_2$), чтобы стоимость требуемой смеси была бы наименьшей, либо сообщить причину, почему это нельзя сделать.

2.42. Число прописью. Дано целое положительное число $N \leqslant 999$. Записать это число в виде числительного

а) русского языка;

б) английского языка.

2.43. О рублях. Дано целое неотрицательное число $N \leq 999$. Просклонять числительное, обозначающее денежную сумму в N рублей, по падежам.

2.44. Покер. Даны пять чисел. Если все они равны друг другу, то вывести число 1; если одинаковы ровно четыре числа, то вывести 2; если равны друг другу три числа и равны друг другу два других числа, то вывести 3; если одинаковы три, то вывести 4; если одинаковы два и два числа, то вывести 5; если одинаковы только два числа, то вывести 6; во всех остальных случаях вывести 7.

3. Цикл с заранее известным числом повторений

А

3.1. Последовательности. Дано целое n $(1 \leqslant n \leqslant 10)$. Найти сумму и произведение элементов последовательности:

а) $\left(\frac{1}{1}\right)^n, \left(\frac{1}{2}\right)^n, \cdots, \left(\frac{1}{n}\right)^n;$

б) $\left(\frac{1}{1}\right)^1, \left(\frac{1}{2}\right)^2, \cdots, \left(\frac{1}{n}\right)^n;$

в) $\left(\frac{1}{1}\right)^n, \left(\frac{1}{2}\right)^{n-1}, \cdots, \left(\frac{1}{n}\right)^1;$

г) $\left(1+\frac{1}{1}\right)^1, \left(1+\frac{1}{2}\right)^2, \cdots, \left(1+\frac{1}{n}\right)^n;$

д) $\left(1+\frac{1}{1^n}\right), \left(1+\frac{1}{2^n}\right), \cdots, \left(1+\frac{1}{n^n}\right);$

е) $\left(\frac{1}{1}\right)^1, \left(\frac{1}{1+2}\right)^2, \cdots, \left(\frac{1}{1+2+\cdots+n}\right)^n;$

ж) $\left(\frac{n}{1}\right)^2, \left(\frac{n-1}{1+2}\right)^3, \cdots, \left(\frac{1}{1+2+\cdots+n}\right)^{n+1};$

з) $\left(\frac{1^n}{\frac{1}{1}}\right), \left(\frac{2^{n-1}}{\frac{1}{1}+\frac{1}{2}}\right), \cdots, \left(\frac{n^1}{\frac{1}{1}+\frac{1}{2}+\cdots+\frac{1}{n}}\right);$

и) $\frac{2^1}{1!+1}, \frac{2^2}{2!+1+2}, \cdots, \frac{2^n}{n!+1+2+\cdots+n};$

к) $\left(\frac{1}{1!}\right), \left(\frac{1}{1!}+\frac{1+2}{2!}\right), \cdots,$
$\left(\frac{1}{1!}+\frac{1+2}{2!}+\cdots+\frac{1+2+\cdots+n}{n!}\right);$

л) $\left(\frac{(-1)^1(n+1)^0}{n!}\right), \left(\frac{(-1)^2(n)^1}{(n-1)!}\right), \cdots, \left(\frac{(-1)^n(2)^{n-1}}{1!}\right);$

м) $\dfrac{1^{n-1}}{\frac{1}{1}}, \dfrac{2^{n-2}}{\frac{1}{1}+\frac{1}{1+2}}, \cdots, \dfrac{n^0}{\frac{1}{1}+\frac{1}{1+2}+\cdots+\frac{1}{1+2+\cdots+n}}$;

н) $\dfrac{1^{n-1}}{1!+n}, \dfrac{1^{n-1}+2^{n-2}}{2!+(n-1)}, \cdots, \dfrac{1^{n-1}+2^{n-2}+\cdots+n^0}{n!+1}$;

о) $\dfrac{(-1)^0}{\frac{(-1)^1}{1}}, \dfrac{(-1)^1}{\frac{(-1)^1}{1}+\frac{(-1)^2}{2}}, \cdots, \dfrac{(-1)^{n-1}}{\frac{(-1)^1}{1}+\frac{(-1)^2}{2}+\cdots+\frac{(-1)^n}{n}}$;

п) $\left(\dfrac{(-1)^0\cdot n}{1+1!}-1\right), \left(\dfrac{(-1)^1\cdot(n-1)}{1+2+2!}-2\right), \cdots,$
$\left(\dfrac{(-1)^{n-1}\cdot 1}{1+2+\cdots+n+n!}-n\right).$

3.2. Схема Горнера. Дано вещественное x, целое $n, n > 0$. Найти значение многочлена n-й степени $P(x)$ в точке x по схеме Горнера

$$P(x) = a_n x^n + a_{n-1}x^{n-1} + \cdots + a_1 x + a_0 = \\ = ((\cdots(a_n x + a_{n-1})x + \cdots)x + a_1)x + a_0$$

Значения коэффициентов a_i $(i = 0, 1, 2, \ldots, n)$ взять равными 1.

3.3. Решить задачу 3.2 для коэффициентов многочлена $a_i = i$; $i = 0, 1, 2, \ldots, n$.

3.4. Решить задачу 3.2 для коэффициентов многочлена $a_i = 2^i$; $i = 0, 1, 2, \ldots, n$.

3.5. Решить задачу 3.2 для коэффициентов многочлена $a_i = i!$; $i = 0, 1, 2, \ldots, n$.

3.6. Решить задачу 3.2 для последовательно вводимых с клавиатуры коэффициентов многочлена a_i; $i = n, n-1, \ldots, 0$.

3.7. Дано целое положительное n. Найти $\underbrace{\sqrt{2+\sqrt{2+\cdots+\sqrt{2}}}}_{n\text{ корней}}$.

3.8. Дано целое положительное n. Найти $\sqrt{1+\sqrt{2+\dots+\sqrt{n}}}$.

3.9. Дано целое положительное n. Найти $\sqrt{2+\sqrt{4+\dots+\sqrt{2n}}}$.

3.10. Дано целое положительное n. Найти
$\sqrt{\dots\sqrt{\sqrt{3n}+3(n-1)}+\dots+3}$.

3.11. Дано целое положительное n, вещественное x. Найти

а) $\sin x+\sin\sin x+\dots+\underbrace{\sin\sin\cdots\sin}_{n} x$;

б) $\sin x+\sin x^2+\dots+\sin x^n$;

в) $\sin x+\sin^2 x+\dots+\sin^n x$;

г) $\sin x+\sin\sin x^2+\dots+\underbrace{\sin\sin\cdots\sin}_{n} x^n$.

3.12. Дано целое положительное n. Найти

$$\cfrac{1}{1+\cfrac{1}{2+\cfrac{1}{\begin{matrix}\ddots\\ n+\cfrac{1}{n+1}\end{matrix}}}}.$$

3.13. Дано целое положительное n. Найти

$$\cfrac{1}{n+\cfrac{1}{(n-1)+\cfrac{1}{\begin{matrix}\ddots\\ 2+\cfrac{1}{1}\end{matrix}}}}.$$

3.14. Дано целое положительное n. Найти

$$\cfrac{1}{2^0 + \cfrac{1}{2^1 + \cfrac{1}{\ddots_{\, 2^n + \frac{1}{2^{n+1}}}}}}.$$

3.15. Таблица значений функции. Вычислить и вывести на экран в виде таблицы значения функции $f(x)$ в точках x_i отрезка $[a;b]$ $(i = 0, 1, \ldots, n)$, где $x_0 = a$, $x_n = b$, остальные точки x_i равномерно распределены внутри отрезка $[a;b]$:

а)

$$f(x) = \begin{cases} ax^2 - bx + 1, & \text{если } x^2 > 1 \text{ и } b < 0; \\ -ax^2 + bx + 1, & \text{если } x^2 \leqslant 1 \text{ и } b < 0; \\ \dfrac{x}{b-a} & \text{в остальных случаях;} \end{cases}$$

б)

$$f(x) = \begin{cases} 2ax^2 + bx + (a - b), & \text{если } (x+1)^2 > 4 \text{ и } a > 0; \\ x^3 - b, & \text{если } (x+1)^2 \leqslant 4 \text{ и } a > 0; \\ \dfrac{a+b}{x^2} & \text{в остальных случаях;} \end{cases}$$

в)

$$f(x) = \begin{cases} \dfrac{a^2 + b^2}{x^2 + 1}, & \text{если } x > a \text{ и } x < \dfrac{a+b}{2}; \\ \dfrac{x^2 + 1}{a^2 + b^2}, & \text{если } x < b \text{ и } x > \dfrac{a+b}{2}; \\ \dfrac{x}{b-a} & \text{в остальных случаях.} \end{cases}$$

3.16. Даны вещественные a, b $(ab \neq 0)$, целое неотрицательное n. Среди чисел вида $a_0 = b \sin a, a_i = b \sin(a \cdot a_{i-1}), i = 1, 2, \ldots, n$ найти наибольшее и наименьшее.

3.17. Дано целое $n > 0$. Среди чисел вида $a_i = i \sin \frac{1}{i}, i = 1, 2, \ldots, n$ найти наибольшее и наименьшее.

3.18. Дано целое $n > 0$, вещественное x. Среди чисел вида $a_i = e^{\cos x^{2i}} \cdot \sin x^{3i}, i = 1, 2, \ldots, n$ найти наибольшее, наименьшее и ближайшее к какому-либо целому.

3.19. Счастливые билеты. Даны целые положительные M и N ($100000 \leqslant M \leqslant N$). Найти количество автобусных билетов с 6-значными номерами от M до N, у которых сумма первых трех десятичных цифр равна сумме трех последних десятичных цифр.

3.20. Числа Фибоначчи. Числовая последовательность a_i $(i = 1, 2, 3\ldots)$, задаваемая по правилам $a_1 = 1,\ a_2 = 1,\ a_i = a_{i-1} + a_{i-2}$ $(i = 3, 4, 5, \ldots)$, называется последовательностью чисел Фибоначчи. Дано целое неотрицательное n. Найти n-е число Фибоначчи.

3.21. Найти количество точек плоскости с целочисленными координатами, попадающими в круг радиусом r с центром в точке (x_c, y_c).

3.22. Найти количество точек плоскости с целочисленными координатами, попадающими в кольцо, образованное окружностями с центром в точке (x_c, y_c) и радиусами r_1 и r_2.

3.23. Найти количество точек плоскости с целочисленными координатами, попадающими в эллипс $\dfrac{x^2}{a^2} + \dfrac{y^2}{b^2} \leqslant 1$.

3.24. Даны вещественные b, c. Найти количество точек плоскости с целочисленными координатами, попадающими в фигуру, ограниченную линиями $y = x^2 + 2bx + c$ и $y = b^2 + c$.

3.25. Коэффициенты бинома Ньютона. Даны целые n, m $(0 \leqslant m \leqslant n)$. Найти коэффициент бинома Ньютона $C_n^m = \frac{n!}{m!(n-m)!}$.

3.26. Таблица истинности. Напечатать таблицу истинности бинарных логических операций *and*, *or*, *xor*.

3.27. Даны натуральные n, m $(2 \leqslant n \leqslant 10, 1 \leqslant m \leqslant 5)$. Напечатать все m-значные натуральные числа в n-ичной системе счисления.

В

3.28. Решить задачу 3.20, используя для хранения чисел Фибоначчи только две переменные.

3.29. Треугольник Паскаля. Дано целое неотрицательное число K. Напечатать «треугольник Паскаля» - таблицу биномиальных коэффициентов C_n^m (см. задачу 3.25) для всех возможных целых значений m и n ($0 \leqslant m \leqslant n \leqslant K$).

3.30. Найти значение коэффициента бинома Ньютона (см. задачу 3.25) за наименьшее количество операций умножения.

3.31. Дано целое $n, 0 \leqslant n \leqslant 27$. Найти количество натуральных чисел, меньших 1000, сумма цифр которых равна n.

3.32. Совершенные числа. Натуральное число называется *совершенным*, если оно равно сумме всех своих делителей, кроме самого себя (например, число 28=1+2+4+7+14 является совершенным). Дано натуральное N, ($N <= 10000$). Найти все совершенные числа, не превосходящие N.

3.33. Даны натуральные N, M ($N \leqslant M \leqslant 1000$). Найти все простые числа p, удовлетворяющие неравенствам $N \leqslant p \leqslant M$.

3.34. Дано натуральное $N \leqslant 1000$. Среди натуральных чисел, не превосходящих N, найти число с наибольшей суммой делителей.

3.35. Дано натуральное $N \leqslant 1000$. Найти все тройки натуральных чисел a, b и c ($a \leqslant b \leqslant c$), удовлетворяющих условию $a^2 + b^2 + c^2 = N$.

3.36. Дано натуральное число n. Напечатать таблицу перевода сантиметров в дюймы, и наоборот, состоящую из n строк.

Точность – два десятичных знака.

Пример.

Введите n=5

Таблица перевода:

cm	inch
1.00	0.39
2.00	0.79
2.54	1.00
3.00	1.18
4.00	1.57.

C

3.37. Пятница, 13-е. Дано натуральное число $m \leqslant 7$. Считая, что 1-е января является m-м по счету днем недели и что год не является високосным, найти количество тринадцатых чисел месяцев года, выпадающих на пятницу.

3.38. С первого числа. Дано натуральное число $m \leqslant 7$. Считая, что 1-е января является m-м по счету днем недели и что год не является високосным, найти, сколько недель в году начинется первого числа.

3.39. Понедельники. Дано натуральное число $m \leqslant 7$. Считая, что 1-е января является m-м по счету днем недели и что год не является високосным, определить, в скольких месяцах года количество понедельников больше количества воскресений.

3.40. Снежинка Коха. Определим плоскую геометрическую фигуру «снежинка» N-го порядка следующим образом: правильный треугольник со стороной a является «снежинкой» первого порядка. На средней трети каждой стороны «снежинки» как на основании построим «наружу» правильный треугольник со стороной втрое меньше исходной стороны. Внешние стороны полученной фигуры образуют «снежинку» второго порядка. Процесс продолжается до N-го построения. Даны вещественное $a > 0$ и целое $N > 0$. Найти площадь и периметр «снежинки» N-го порядка, если длина стороны «снежинки» первого порядка равна a.

3.41. Решить задачу 3.40 для квадратов.

3.42. Перестановки. Дано натуральное n. Найти все перестановки первых n натуральных чисел.

Пример.

Введите n=*3*
Все перестановки из 3 чисел:
1 2 3
1 3 2
2 1 3
2 3 1
3 1 2
3 2 1

3.43. Сочетания. Даны натуральные n и m ($n \geqslant m$). Найти все различные сочетания первых n натуральных чисел, взятых по m штук.

Пример.

Введите n=*3*
Введите m=*2*
Все сочетания из 3 чисел по 2:
1 2
1 3
2 3

3.44. Размещения. Даны натуральные n и m ($n \geqslant m$). Найти все различные размещения n первых натуральных чисел, взятых по m штук.

3.45. Телефонный номер называется «шахматным», если его цифры набираются на телефонном кнопочном номеронабирателе ходом шахматного коня. Написать программу, подсчитывающую, сколько можно набрать различных семизначных «шахматных» номеров, начинающихся с заданной цифры.

4. Цикл с условием

А

4.1. Дано вещественное a. Найти все члены последовательности $\frac{1}{1}, \frac{1}{2}, \frac{1}{3}, \cdots$, большие a.

4.2. Дано вещественное a. Найти все числа Фибоначчи (см. задачу 3.20), не превосходящие a.

4.3. Дано вещественное b. Среди элементов последовательности указать первый, больший b:

а) $\frac{1}{1}, \frac{1}{1}+\frac{1}{2}, \frac{1}{1}+\frac{1}{2}+\frac{1}{3}, \cdots;$

б) $\frac{1}{1}, \frac{1}{1}+\frac{1}{3}, \frac{1}{1}+\frac{1}{3}+\frac{1}{5}, \cdots;$

в) $\frac{1}{2}, \frac{1}{2}+\frac{2}{3}, \frac{1}{2}+\frac{2}{3}+\frac{3}{4}, \cdots.$

4.4. Вычисление квадратного корня. Даны положительные x, d. Найти номер и значение первого члена последовательности a_i, построенной по правилу $a_1 = \frac{x+1}{2}, a_i = \frac{1}{2}\left(a_{i-1} + \frac{x}{a_{i-1}}\right), i = 2, 3, \ldots$, квадрат которого отличается от числа x не более чем на d (Формула *Герона Александрийского*).

4.5. Дано положительное d. Найти номер и значение первого члена последовательности a_i, построенной по правилу $a_i = i \sin\frac{1}{i}$ $i = 1, 2, \ldots$, который отличается от 1 не более чем на d. Проверить для значений $d = 10^{-1}, 10^{-2}, 10^{-3}, \ldots, 10^{-6}$.

4.6. Среди элементов последовательности найти первый элемент, отличающийся от предыдущего не более чем на $\varepsilon = 10^{-6}$:

а) $\frac{1}{1}, \frac{1}{4}, \frac{1}{9}, \frac{1}{16}, \cdots;$

б) $\frac{1}{1}, \frac{1}{1}+\frac{1}{4}, \frac{1}{1}+\frac{1}{4}+\frac{1}{9}, \frac{1}{1}+\frac{1}{4}+\frac{1}{9}+\frac{1}{16}, \cdots;$

в) $\frac{1}{1}, \frac{1}{1}-\frac{1}{2}, \frac{1}{1}-\frac{1}{2}+\frac{1}{3}, \frac{1}{1}-\frac{1}{2}+\frac{1}{3}-\frac{1}{4}, \cdots.$

4.7. Дано вещественное $b > 0$. Среди членов последовательности a_i, построенной по правилу: $a_1 = 2, a_i = a_{i-1}^2 - 1, i = 2, 3, \ldots$, получить все те элементы, которые меньше b.

4.8. Дано вещественное $b > 0$. Последовательность a_i построена по правилу: $a_1 = b, a_i = a_{i-1} - \frac{1}{\sqrt{i}}, i = 2, 3, \ldots$. Найти номер и значение первого отрицательного члена последовательности.

4.9. Дано вещественное $b < 1$. Найти все числа последовательности a_i, построенной по правилу $a_1 = \frac{1}{2}, a_i = a_{i-1} + \frac{1}{2^i}, i = 2, 3, \ldots$, меньшие b.

4.10. Дано вещественное $b \geqslant 1$. Найти все члены последовательности a_i, построенной по правилу $a_1 = 1, a_i = a_{i-1} + \frac{1}{a_{i-1}}, i = 2, 3, \ldots$, меньшие b.

4.11. Вычислить значение функции $f(x)$в точке x, $(|x| < 1)$ с помощью разложения её в бесконечный ряд

$$f(x) = \sum_{i=0}^{\infty} a_i$$

с точностью $\varepsilon > 0$. Требуемая точность считается достигнутой, если найдена сумма некоторого количества первых слагаемых, а последующее слагаемое оказалось по модулю меньше, чем ε. Полученный результат сравнить со значением, найденным с использованием соответствующих стандартных функций:

а) $e^x = \sum\limits_{i=0}^{\infty} \frac{x^i}{i!};$

б) $\frac{e^x - e^{-x}}{2} = \sum\limits_{i=0}^{\infty} \frac{x^{2i+1}}{(2i+1)!};$

в) $\dfrac{e^x+e^{-x}}{2} = \sum\limits_{i=0}^{\infty} \dfrac{x^{2i}}{(2i)!}$;

г) $\cos x = \sum\limits_{i=0}^{\infty} \dfrac{(-1)^i x^{2i}}{(2i)!}$;

д) $\sin x = \sum\limits_{i=1}^{\infty} \dfrac{(-1)^{i-1} x^{2i-1}}{(2i-1)!}$;

е) $\ln(x+1) = \sum\limits_{i=1}^{\infty} \dfrac{(-1)^{i-1} x^{i}}{i}$;

ж) $arctg\, x = \sum\limits_{i=0}^{\infty} \dfrac{(-1)^i x^{2i+1}}{(2i+1)}$;

з) $\sin x - \cos x + 1 = \sum\limits_{i=1}^{\infty} \dfrac{(-1)^{i-1} x^{2i-1}(2i+x)}{(2i)!}$;

и) $\dfrac{1-\cos x - x\sin x}{x^2} + \dfrac{1}{2} = \sum\limits_{i=1}^{\infty} \dfrac{(-1)^{i+1}(2i+1)x^{2i}}{(2i+2)!}$;

к) $\dfrac{\cos x - 1}{x^2} + \dfrac{1}{2} = \sum\limits_{i=1}^{\infty} \dfrac{(-1)^{i+1} x^{2i}}{(2i+2)!}$;

л) $2 - e^{-x^2} - \cos x = \sum\limits_{i=1}^{\infty} (-1)^{i+1} x^{2i} \left(\dfrac{1}{i!} + \dfrac{1}{(2i)!}\right)$;

м) $(1+2x^2)e^{x^2} = \sum\limits_{i=0}^{\infty} \dfrac{2i+1}{i!} x^{2i}$;

н) $xe^{-x} - e^{-x} + 1 = \sum\limits_{i=1}^{\infty} (-1)^{i+1} \dfrac{(i+1)x^i}{i!}$;

о) $\dfrac{x-\sin x}{x^2} = \sum\limits_{i=1}^{\infty} (-1)^{i+1} \dfrac{x^{2i-1}}{(2i+1)!}$;

п) $2\sin^2 x = \sum\limits_{i=1}^{\infty} (-1)^{i+1} \dfrac{(2x)^{2i}}{(2i)!}$;

р) $\cos\sqrt{x} - e^{-x} = \sum\limits_{i=1}^{\infty} (-1)^{i+1} x^i \left(\dfrac{1}{i!} - \dfrac{1}{(2i)!}\right)$;

с) $\sin x^2 - x^2 \cos x^2 = \sum\limits_{i=1}^{\infty}(-1)^{i+1}\dfrac{(2i)x^{4i+1}}{(2i+1)!}$;

т) $2x - xe^{-x^2} - \sin x = \sum\limits_{i=1}^{\infty}(-1)^{i+1}x^{2i+1}\left(\dfrac{1}{i!} + \dfrac{1}{(2i+1)!}\right)$;

у) $x + 1 - \dfrac{\pi^2}{12} - \ln(x+1) = \sum\limits_{i=1}^{\infty}(-1)^{i+1}\dfrac{(i+1)x^{i+1}+1}{(i+1)^2}$;

ф) $2x \cdot arctg\, x - 2\ln\sqrt{1+x^2} = \sum\limits_{i=1}^{\infty}(-1)^{i+1}\dfrac{x^{2i}}{i(2i-1)}$.

B

4.12. Дано натуральное число $N \leqslant 2000000000$. Представить N в виде суммы квадратов натуральных чисел, содержащей наименьшее число слагаемых.

4.13. Дано натуральное число $N \leqslant 2000000000$. Представить N в виде суммы факториалов натуральных чисел, содержащей наименьшее число слагаемых.

4.14. Алгоритм Евклида. Даны целые неотрицательные a, b. Найти наибольший общий делитель (НОД) a и b. Для нахождения НОД a и b можно воспользоваться алгоритмом Евклида:

$$\text{НОД}(a,b) = \begin{cases} a, & \text{если } a = b; \\ \text{НОД}(a-b, b), & \text{если } a > b; \\ \text{НОД}(a, b-a), & \text{если } a < b. \end{cases}$$

4.15. Алгоритм Евклида-2. Решить задачу 4.14, используя другой вариант алгоритма Евклида:

$$\text{НОД}(a,b) = \begin{cases} a, & \text{если } b = 0; \\ \text{НОД}(b, r), & \text{в противном случае;} \\ & r\text{– остаток от деления } a \text{ на } b. \end{cases}$$

4.16. Дано натуральное число N. Найти все натуральные числа, меньшие N и взаимно простые с ним.

4.17. Дано натуральное число N. Разложить N в произведение простых сомножителей.

4.18. Из цифр записи двух данных натуральных чисел составить наибольшее возможное число, сохраняя первоначальную последовательность цифр.

4.19. Даны натуральные числа a, b, c, d. Найти сумму $\frac{a}{b} + \frac{c}{d}$ в виде правильной несократимой дроби с выделенной целой частью.

Пример.

$$\frac{5}{3} + \frac{4}{6} = 2\frac{1}{3}.$$

4.20. Дано целое неотрицательное N. Напечатать это число в двоичной, восьмеричной и шестнадцатеричной системах счисления.

4.21. Усталый путник. Путнику необходимо преодолеть расстояние s километров. Его скорость в начале пути равна v км/ч. Через каждый час из-за усталости его скорость уменьшается в m раз ($m \geqslant 1$). Преодолеет ли путник это расстояние и, если преодолеет, то за какое время?

В задачах 4.22-4.35 применяются ограничения: натуральные числа, если особо не оговорено, не превышают 2 миллиардов; нельзя использовать массивы, строковые переменные.

4.22. Дано целое число N. Найти произведение наибольшей и наименьшей цифр числа N.

4.23. Даны натуральные числа N и M ($M \leqslant 10$). Найти сумму первых M цифр числа N. Если цифр в N меньше, чем M, то считать недостающие цифры равными 0.

4.24. Даны натуральные числа N и M ($0 \leqslant M \leqslant 9$). Найти число вхождений цифры M в число N.

4.25. Дано натуральное число N. Найти число, получаемое из N выбрасыванием всех нечетных цифр.

4.26. Дано натуральное число N ($N \leqslant 40000$). Найти число, получаемое из N дублированием всех четных цифр.

C

4.27. Дано натуральное число N. Указать наиболее часто встречающиеся цифры числа N.

4.28. Дано натуральное число N. Поменять порядок цифр числа N на обратный.

4.29. Дано натуральное число N. Определить, является ли это число палиндромом.

4.30. Дано натуральное число N. Указать самую длинную неубывающую подпоследовательность цифр числа N.

4.31. Дано натуральное число N. Найти основание системы счисления $M \leqslant 10$, в которой запись числа N содержит наибольшее число единиц.

4.32. Дано натуральное число N. Указать N-ю цифру последовательности $12345678910111213\ldots$, в которой выписаны подряд все натуральные числа.

4.33. Дано натуральное число N. Указать N-ю цифру последовательности $149162536\ldots$, в которой выписаны подряд квадраты всех натуральных чисел.

4.34. Дано натуральное число N. Указать N-ю цифру последовательности $112358132134\ldots$, в которой выписаны подряд все числа Фибоначчи (см. задачу 3.20).

4.35. Дано натуральное число N. Найти число, получаемое из N выбрасыванием рядом стоящих значащих цифр, отличающихся на 1.

Пример.

Введите N=*134307*
Полученное число равно 107.

4.36. Разрезать. Какое минимальное число прямолинейных разрезов нужно сделать, чтобы разрезать прямоугольную пластину размером $m \times n$ на равные квадраты максимальной площади?

4.37. Дано вещественное неотрицательное N. Напечатать это число в двоичной, восьмеричной и шестнадцатеричной системах счисления.

4.38. Двоичные числа Фибоначчи. Последовательность
$0111011101100011011010 1\ldots$
сформирована из цифр последовательных чисел Фибоначчи (см. задачу 3.20), записанных в двоичной системе счисления. Дано натуральное $N (N \leqslant 6)$. Найти количество нулей и единиц этой последовательности вплоть до первой встретившейся группы из N нулей или единиц.

4.39. Простые соседи. Сформулируем тезис: «существует такое натуральное число N, что разность между любыми двумя соседними простыми числами, не превосходящими N, не больше 255, а для некоторых соседних простых чисел, больших N, это условие нарушается». Требуется либо найти N, либо попробовать доказать, что такого N не существует.
Кстати, если этот тезис верен, то для представления любого конечного подмножества последовательных простых чисел можно использовать массив байтов.

4.40. Треугольник. Дан треугольник из натуральных чисел: в первой строке – одно число, во второй – два и т.д. (числа не превосходят 100). Двигаемся от вершины треугольника вниз, каждый шаг может быть от числа к любому из его двух соседей внизу. Останавливаемся на основании треугольника. Необходимо построить такой порядок прохода по треугольнику, чтобы сумма пройденных чисел была бы минимальна.

Пример.

Треугольник:

$$\begin{array}{ccccccc} & & & 5 & & & \\ & & 4 & & 3 & & \\ & 2 & & 4 & & 5 & \\ 6 & & 8 & & 2 & & 4 \end{array}$$

Порядок прохода: 5, 3, 4, 2,

при этом наименьшая сумма = 14.

4.41. Обобщение последовательности Фибоначчи. Построим обобщение последовательности чисел Фибоначчи следующим образом: последовательность

$$F^N(1), F^N(2), \ldots, F^N(k), \ldots$$

назовём последовательностью чисел N-Фибоначчи ($N \geqslant 2$– целое число), где

$$F^N(k) = \begin{cases} 1, & k \leqslant N \\ F^N(k-N) + F^N(k-N+1) + \cdots + F(k-1) & k > N \end{cases}$$

Таким образом, последовательность 2-Фибоначчи
1 1 2 3 5 8 13 21 34 56 89... —
это обычные числа Фибоначчи, 3-Фибоначчи —
1 1 1 3 5 9 17 31 57 ...
Дано натуральное число A. Найти наименьшее целое $N \geqslant 2$, чтобы A было бы одним из чисел последовательности N-Фибоначчи.

Пример.

Введите A= *57*

Ответ: N=3

Вопрос: всегда ли существует N для произвольного натурального A?

4.42. Числа типа Фибоначчи. Определим последовательность чисел типа Фибоначчи следующим образом:
$u_1 = a$,
$u_2 = b$,
$u_k = u_{k-2} + u_{k-1}$, если $k > 2$,
где a и b — некоторые целые числа. Дано целое число c. Требуется найти такие a и b, чтобы c было членом последовательности типа Фибоначчи и при этом сумма $|a| + |b|$ была минимальна.

5. Линейные массивы

Для решения каждой из задач этого раздела необходимо написать программу, использующую для хранения вещественных значений линейный (одномерный) массив или массивы. Количество элементов массива взять равным некоторой константе n (например, $n = 10$). Заполнять массив следует случайными значениями соответствующего типа (если не указано иначе).

A

5.1. Определить индексы и значения наибольших и наименьших по модулю элементов одномерного массива.

5.2. Найти количество неотрицательных и произведение положительных элементов массива.

5.3. Найти количество элементов массива, модуль разности между которыми и средним арифметическим всех положительных элементов массива не превосходит 1.

5.4. Определить наибольшую по величине разность между соседними элементами массива и индексы всех пар таких элементов.

5.5. Найти среднее арифметическое всех неотрицательных элементов массива.

5.6. Присвоить каждому элементу массива значение следующего элемента, а последнему элементу присвоить значение первого элемента.

5.7. Найти среднее арифметическое элементов массива, начиная с первого элемента и заканчивая последним из всех наибольших элементов.

5.8. Среди отрицательных элементов массива найти такие, значение которых максимально.

5.9. Увеличить отрицательные элементы массива на среднее арифметическое неотрицательных элементов.

5.10. Найти сумму элементов массива, отличающихся от среднего арифметического всех элементов с положительными значениями и нечетными индексами не более чем на 2.

5.11. Дан линейный вещественный массив a. Найти:
а) $a_1^2 + a_2^2 + \dots + a_n^2$;
б) $a_1a_n + a_2a_{n-1} + \dots + a_na_1$;
в) $a_1a_2 + a_2a_3 + \dots + a_{n-1}a_n$;
г) $a_1a_2 + a_3a_4 + \dots + a_{n-1}a_n$, (n–четное);
д) $a_1 - a_2 + a_3 - \dots \pm a_n$;
е) $(a_1 + a_2)(a_2 + a_3) \dots (a_{n-1} + a_n)$;
ж) $(a_1 + a_2)(a_3 + a_4) \dots (a_{n-1} + a_n)$, (n–четное);
з) $(a_1 - a_n)(a_2 - a_{n-1}) \dots (a_n - a_1)$;
и) $a_1(a_1 + a_2)(a_1 + a_2 + a_3) \dots (a_1 + a_2 \dots + a_n)$;
к) $a_1 + a_1a_2 + a_1a_2a_3 + \dots + a_1a_2 \cdot \dots \cdot a_n$;
л) $a_1^2 + a_4^2 + a_9^2 + \dots + a_{k^2}^2 + \dots$;
м) $a_1x + a_2x^2 + \dots + a_nx^n$, ($x$ дано);
н) $x^{a_1} + x^{a_2} + \dots + x^{a_n}$, ($x > 0$ дано).

5.12. Даны координаты двух n-мерных векторов $A(a_1; a_2; \dots; a_n)$, $B(b_1; b_2; \dots; b_n)$. Найти:
а) сумму и разность векторов: $A \pm B = C, C(a_1 \pm b_1; \dots; a_n \pm$

b_n);

б) длины векторов: $|A| = \sqrt{\sum\limits_{i=1}^{n} a_i^2}$;

в) скалярное произведение векторов: $A \cdot B = \sum\limits_{i=1}^{n} a_i \cdot b_i$;

г) угол между векторами: $\cos\widehat{(A, B)} = \frac{A \cdot B}{|A| \cdot |B|}$.

5.13. Элементы линейного массива сдвинуть циклически на:

а) одну позицию вправо;

б) две позиции влево;

в) m позиций вправо.

5.14. Дан линейный вещественный массив a. «Зеркально перевернуть» элементы массива, то есть первый элемент поменять значениями с последним, второй – с предпоследним и т.д.

5.15. Дан линейный целочисленный массив a. «Зеркально перевернуть» часть элементов массива, расположенных между первым из всех наименьших и последним из всех наибольших элементов массива.

5.16. Дан линейный целочисленный массив a размером n. Построить массив b размером не более n из различных значений элементов массива в порядке убывания их вхождений в a. В случае, если два разных элемента входят в a одинаковое число раз, разместить их в b в порядке возрастания.

5.17. Даны значения элементов двух массивов x и y размером n, упорядоченные по неубыванию. Объединить значения этих массивов в новый массив z вдвое большего размера так, чтобы они также располагались по неубыванию.

5.18. Все отрицательные зачения линейного вещественного массива переместить в его начало, сохраняя взаимное расположение элементов.

5.19. Даны вещественные значения $a_1, a_2, \ldots, a_n, x$. Найти

$$\sum_{i=1}^{n} \frac{(a_1 - x)(a_2 - x)\cdots(a_{i-1} - x)(a_{i+1} - x)\cdots(a_n - x)}{(a_1 - a_i)(a_2 - a_i)\cdots(a_{i-1} - a_i)(a_{i+1} - a_i)\cdots(a_n - a_i)}.$$

5.20. Дан линейный вещественный массив a. Найти самую длинную неубывающую подпоследовательность последовательности значений $a_1, a_2, \ldots, a_n$.

5.21. Даны коэффициенты многочлена $P(x) = p_n x^n + p_{n-1}x^{n-1} + \cdots + p_1 x + p_0$. Найти

а) для данного x значение $P(x)$;

б) для данных x, a коэффициенты и значение многочлена $Q(x) = (x - a)P(x)$;

в) для данного x коэффициенты и значение производной многочлена $P(x)$: $R(x) = (P(x))'$;

г) для данных коэффициентов многочлена $S(x) = s_m x^m + s_{m-1}x^{m-1} + \cdots + s_1 x + s_0$ коэффициенты произведения многочленов $P(x) \cdot S(x)$;

д) для данных коэффициентов многочлена $T(x) = t_m x^m + t_{m-1}x^{m-1} + \cdots + t_1 x + t_0$, $1 \leqslant m \leqslant 5$ коэффициенты многочлена $P(T(x))$.

5.22. Дано целое $k\,(2 \leqslant k \leqslant 20)$. Найти коэффициенты k-го многочлена Чебышева. Последовательность $T_i(x)$ *многочленов Чебышева* задается формулами: $T_0(x) = 1; T_1(x) = x; T_i(x) = 2xT_{i-1}(x) - T_{i-2}(x), i = 2, 3, 4, \ldots$.

В

5.23. Даны значения двух целочисленных массивов x и y размером n. Рассматривая массивы как конечные множества целых чисел, построить массив z размером не более $2n$, где

а) $z = x \cap y$ (пересечение множеств);

б) $z = x \cup y$ (объединение множеств);

в) $z = x \setminus y$ (разность множеств);

г) $z = x \triangle y$ (симметрическая разность множеств[1]).

5.24. Даны значения двух вещественных массивов x и y размером n. Рассматривая пары значений $(x_1; y_1), \ldots (x_n; y_n)$ как координаты точек плоскости, найти:

а) номера двух наиболее удаленных точек;

б) номера трех точек, которые являются вершинами треугольника с наибольшей площадью;

в) номер такой точки и величину r, чтобы окружность с центром в этой точке содержала все n точек, и при этом радиус r ее был наименьшим среди радиусов всех таких окружностей.

5.25. Дан линейный вещественный массив a_1, a_2,…,a_n. Определить максимальное количество подряд идущих положительных элементов массива, не прерываемых ни нулями, ни отрицательными элементами, вывести на экран указанную последовательность элементов.

5.26. Дана последовательность вещественных чисел a_1, a_2,…,a_n. Указать самую длинную неубывающую подпоследовательность данной последовательности.

В задачах 5.27–5.30 требуется провести *сортировку* линейного числового массива по неубыванию. Это значит так переставить элементы массива a, чтобы для всех пар соседних элементов выполнялось неравенство $a_k \leqslant a_{k+1}$.

5.27. Сортировка выбором. Среди элементов массива $a_1, a_2, \ldots,\ a_n$ выбирается наименьший и меняется местами с элементом a_1. Затем этот алгоритм применяется к элементам $a_2, a_3, \ldots, a_n$ и т.д. до тех пор, пока не останется последний элемент массива.

5.28. Сортировка вставками. Алгоритм представляет собой последовательное применение $n-1$ раз следующего шага

[1] Симметрическая разность определяется так: $A \triangle B = (A \cup B) \setminus (A \cap B)$.

(для k от 1 до $n-1$): допустим, первые k элементов $a_1, a_2, \ldots a_k$ уже отсортированы; для элемента a_{k+1} находится подходящее место среди первых k элементов, и он вставляется на это место, при этом элементы, находящиеся правее места вставки, вплоть до a_k, сдвигаются на одну позицию вправо.

5.29. Сортировка обменом (пузырьковая сортировка). Алгоритм представляет собой последовательное применение $n-1$ раз следующего шага (для k от n до 2): последовательно сравниваются пары соседних элементов a_i и a_{i+1} (i от 1 до $k-1$), если элементы в паре неупорядочены (то есть $a_i > a_{i+1}$), то они обмениваются значениями.

5.30. Сортировка подсчетом. Для этого алгоритма требуется завести еще один массив – массив счетчиков. Для каждого элемента массива требуется подсчитать количество других элементов, не превосходящих его по значению. В случае равенства значений двух элементов увеличивается счетчик того элемента, индекс которого больше. Таким образом вычисленное значение счетчика каждого элемента (увеличенное на единицу) будет являться индексом его нового положения в отсортированном массиве.

C

5.31. Решить задачу 5.29 пузырьковой сортировки с учетом следующего соображения: часть последних элементов массива в какой-то момент могут уже находиться на своих местах и сортировать их не требуется. Для того чтобы не делать лишние сравнения элементов в этом случае, полезно запомнить место последнего обмена значений элементов и дальше этого места на следующем шаге сортировки не двигаться.

5.32. Отсортировать элементы массива, имеющие нечетные индексы, по неубыванию, а четные индексы – по невозрастанию. Не использовать дополнительных массивов. Применить метод:

а) сортировки выбором;
б) сортировки вставками;
в) пузырьковой сортировки.

5.33. Отсортировать отрицательные элементы массива по невозрастанию, не затрагивая при этом остальные элементы, а неотрицательные – по неубыванию, не затрагивая остальные. Например, массив
3 -2 4 -3 -1 0
после сортировки должен выглядеть так:
0 -1 3 -2 -3 4.
Дополнительные массивы использовать нельзя. Применить метод:

а) сортировки выбором;
б) сортировки вставками;
в) пузырьковой сортировки.

5.34. Прогрессии в массиве. В линейном вещественном массиве найти самую длинную цепочку подряд идущих элементов, являющихся членами арифметической или геометрической прогрессии.

6. Матрицы

Для решения каждой из задач этого раздела необходимо написать программу, использующую для хранения вещественных значений двумерные массивы (матрицы). Заполнять матрицу следует случайными значениями соответствующего типа (если не указано иначе) .

А

6.1. Заполнить целочисленную квадратную матрицу A размером 10×10 следующим образом:

а) каждый элемент равен номеру строки, в которой он находится;

б) каждый элемент равен номеру столбца, в котором он находится;

в) каждый элемент равен сумме его индексов;

г) каждый элемент равен наибольшему из его индексов;

д) каждый элемент главной диагонали равен 1, остальные элементы равны 0;

е) каждый элемент выше главной диагонали равен сумме индексов, на главной диагонали – номеру строки, а ниже – 0.

ж) каждый элемент на побочной диагонали равен 0, все остальные элементы равны 1;

з) каждый элемент выше побочной диагонали равен наибольшему из его индексов, на побочной диагонали – номеру столбца, а ниже – 1;

и) каждый элемент в нечетной строке равен номеру столбца, а в четном – номеру строки;

к) каждый элемент в нечетном столбце равен номеру строки, а в четном – номеру столбца.

6.2. Дана вещественная матрица A размером 5×4. Найти наименьшую из сумм элементов строк матрицы и сумму наименьших элементов столбцов.

6.3. Дана вещественная матрица A размером 6×7. Найти наибольшее значение среди сумм модулей элементов столбцов матрицы и сумму наибольших значений строк.

6.4. Дана вещественная матрица A размером 4×5. Найти наименьшее значение среди положительных минимумов строк и вычесть его из каждого элемента матрицы.

6.5. Дана вещественная матрица A размером 4×5. Найти сумму наибольших значений модулей элементов столбцов матрицы и прибавить ее к каждому элементу матрицы.

6.6. Дана вещественная квадратная матрица A порядка 7. Произвести «зеркальный разворот» строки матрицы, в которой находится наименьший среди элементов главной диагонали (под «зеркальным разворотом» строки матрицы будем понимать обмен значениями первого и последнего элементов строки, второго и предпоследнего и т.д.).

6.7. Дана вещественная квадратная матрица A порядка 8. Произвести «зеркальный разворот» столбца матрицы, в которой находится наибольший элемент среди элементов побочной диагонали (см. задачу 6.6).

6.8. Дана вещественная матрица A размером 8×6. Поменять местами строку, содержащую наибольший элемент матрицы со строкой, содержащей наименьший элемент.

6.9. Дана вещественная квадратная матрицы A порядка n. Считая, что $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n$, найти:

а) произведение матрицы на вещественное число $C = kA$, где $c_{ij} = ka_{ij}$, k – вещественное число;

б) транспонированную матрицу A^T, где $a_{ij}^T = a_{ji}$;

в) произведение матрицы на вектор $y = Ax$, где $y_i = \sum\limits_{k=1}^{n} a_{ik}x_k$, x, y – вещественные векторы.

6.10. Даны две вещественные квадратные матрицы A и B порядка n. Считая, что $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n$, найти:

а) сумму матриц $C = A + B$, где $c_{ij} = a_{ij} + b_{ij}$;

б) разность матриц $C = A - B$, где $c_{ij} = a_{ij} - b_{ij}$;

в) произведение матриц $C = AB$, где $c_{ij} = \sum\limits_{k=1}^{n} a_{ik}b_{kj}$.

6.11. Комплексная матрица Z задается в виде пары вещественных матриц A, B: $Z = A + Bi$. Дана квадратная комплексная матрица Z. Считая, что $1 \leqslant i \leqslant n, 1 \leqslant j \leqslant n$, найти:

а) произведение матрицы на комплексное число zZ;

б) транспонированную матрицу Z^T;

в) произведение матрицы на комплексный вектор Zx, где

$x = a + bi; a, b$ – вещественные векторы.

6.12. Комплексная матрица Z задается в виде пары вещественных матриц A, B: $Z = A + Bi$. Даны квадратные комплексные матрицы Z_1, Z_2. Найти:

а) сумму матриц $Z_1 + Z_2$;
б) разность матриц $Z_1 - Z_2$;
в) произведение матриц $Z_1 Z_2$.

6.13. Дана целочисленная матрица A размером $m \times n$. Найти индексы тех строк матрицы, которые являются палиндромами (палиндром – это строка, которая читается одинаково с начала и с конца).

6.14. Даны значения элементов вещественного вектора $x_1, x_2, \ldots, x_n$. Заполнить квадратную матрицу порядка n следующим образом:

а) $\begin{pmatrix} 1 & 1 & \cdots & 1 \\ x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ x_1^{n-1} & x_2^{n-1} & \cdots & x_n^{n-1} \end{pmatrix}$ б) $\begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_1^2 & x_2^2 & \cdots & x_n^2 \\ \cdots & \cdots & \cdots & \cdots \\ x_1^n & x_2^n & \cdots & x_n^n \end{pmatrix}$

в) $\begin{pmatrix} x_1 & x_1^2 & \cdots & x_1^n \\ x_2 & x_2^2 & \cdots & x_2^n \\ \cdots & \cdots & \cdots & \cdots \\ x_n & x_n^2 & \cdots & x_n^n \end{pmatrix}$ г) $\begin{pmatrix} x_1 & x_2 & \cdots & x_n \\ x_n^2 & x_{n-1}^2 & \cdots & x_1^2 \\ x_1^3 & x_2^3 & \cdots & x_n^3 \\ x_n^4 & x_{n-1}^4 & \cdots & x_1^4 \\ \cdots & \cdots & \cdots & \cdots \end{pmatrix}$

B

6.15. Дана вещественная квадратная матрица A порядка n. Упорядочить строки матрицы по неубыванию:

а) первых элементов строк;

б) элементов главной диагонали;
в) элементов побочной диагонали;
г) средних арифметических элементов строк;
д) максимальных элементов строк.

6.16. Дана вещественная матрица A размером $m \times n$. Обозначим $A'(i, j)$ – верхний левый угол матрицы A до i-й строки и j-го столбца (подматрица). Каждому элементу исходной матрицы a_{ij} присвоить значение минимального элемента среди элементов $A'(i, j)$. Ограничение: разрешается в программе использовать единственную матрицу.

6.17. Дана вещественная матрица A размером $m \times n$. Обозначим $A'(i, j)$ – верхний левый угол матрицы A до i-й строки и j-го столбца (подматрица). Каждому элементу исходной матрицы a_{ij} присвоить значение суммы всех элементов $A'(i, j)$. Ограничение: разрешается в программе использовать единственную матрицу.

6.18. Дана целочисленная квадратная матрица A порядка n. Произвести разворот матрицы на 90° против часовой стрелки, то есть элементу a_{11} присвоить значение a_{n1}, элементу a_{n1} - значение a_{nn} и так далее для всех элементов матрицы. Ограничение: разрешается в программе использовать единственную матрицу.

6.19. Седловые точки. Элемент матрицы называется седловой точкой, если он является наименьшим в строке и наибольшим в столбце или, наоборот, наибольшим в строке и наименьшим в столбце. Для данной вещественной матрицы A размером $m \times n$ указать индексы всех седловых точек.

6.20. *Соседом* элемента a_{ij} матрицы называется другой элемент a_{lk} этой же матрицы, если каждый из его индексов l и k отличается от соответственно i и j не более чем на 1. Дана вещественная матрица размером $m \times n$. Построить матрицу B такого же размера, чтобы каждый элемент b_{ij} этой матрицы был равен наименьшему значению среди соседей элемента a_{ij}.

6.21. Дана вещественная матрица A размером $m \times n$. Обозначим M_{ij} матрицу размером $(m-1) \times (n-1)$, получаемую из матрицы A выбрасыванием i-й строки и j-го столбца. Построить матрицу B, где каждый элемент b_{ij} равен наименьшему из элементов матрицы M_{ij}.

6.22. Магический квадрат. Дана целочисленная квадратная матрица A порядка n. Проверить, является ли эта матрица магическим квадратом, то есть такой, в которой суммы элементов во всех строках и во всех столбцах совпадают.

6.23. Латинский квадрат. Дана целочисленная квадратная матрица A порядка n. Проверить, является ли эта матрица латинским квадратом, то есть такой, в которой каждая строка и каждый столбец содержат все числа от 1 до n.

C

6.24. Обратный ход алгоритма Гаусса. Даны коэффициенты и правые части «верхнетреугольной» системы линейных уравнений порядка n. Решить систему

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n &= b_1; \\ \qquad\quad a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n &= b_2; \\ \qquad\qquad\qquad a_{33}x_3 + \cdots + a_{3n}x_n &= b_3; \\ \qquad\qquad\qquad\qquad\qquad\qquad \cdots & \\ \qquad\qquad\qquad\qquad\qquad\quad a_{nn}x_n &= b_n. \end{cases}$$

6.25. Алгоритм Гаусса. Даны коэффициенты и правые части системы линейных уравнений порядка n. Решить систему, предварительно приведя ее к «верхнетреугольному» виду с помощью эквивалентных линейных преобразований (к любому уравнению можно прибавить любое другое уравнение, умноженное на число, отличное от нуля; уравнения можно менять местами в системе):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + \cdots + a_{1n}x_n & = b_1; \\ a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + \cdots + a_{2n}x_n & = b_2; \\ \cdots & \\ a_{n1}x_1 + a_{n2}x_2 + a_{n3}x_3 + \cdots + a_{nn}x_n & = b_n. \end{cases}$$

С помощью i-го уравнения нужно обнулить коэффициенты перед x_i в уравнениях с номерами $i+1, i+2, \ldots, n$. Для этого из каждого j-го уравнения нужно вычесть i-е уравнение, умноженное на $\frac{a_{ji}}{a_{ii}}$.

6.26. Алгоритм Гаусса с выбором главного элемента. Это разновидность алгоритма Гаусса (задача 6.25), в котором перед обнулением i-го столбца коэффициентов необходимо выбрать уравнение, которое будет обнулять эти коэффициенты. В качестве «обнуляющего» уравнения выбирают такое уравнение, в котором коэффициент перед x_i является наибольшим по модулю в уравнениях с номерами $i, i+1, \ldots, n$.

6.27. Алгоритм Жордана. Алгоритм Жордана похож на алгоритм Гаусса (задача 6.25), но во время прямого хода с помощью i-го уравнения обнуляются не только соответствующие коэффициенты в $(i+1)$-м, $(i+2)$-м $\ldots$, n -м уравнениях, но и в 1-м 2-м, $\ldots$, $(i-1)$-м уравнениях. В этом случае обратного хода не требуется.

6.28. Дана вещественная квадратная матрица порядка n. Найти определитель матрицы методом Гаусса (см. задачу 6.25), то есть привести вначале матрицу к «верхнетреугольному» виду линейными преобразованиями строк. Определитель «верхнетреугольной» (и исходной) матрицы равен произведению элементов главной диагонали.

6.29. Дана вещественная квадратная матрица порядка n. Найти обратную ей матрицу или установить, что такой не суще-

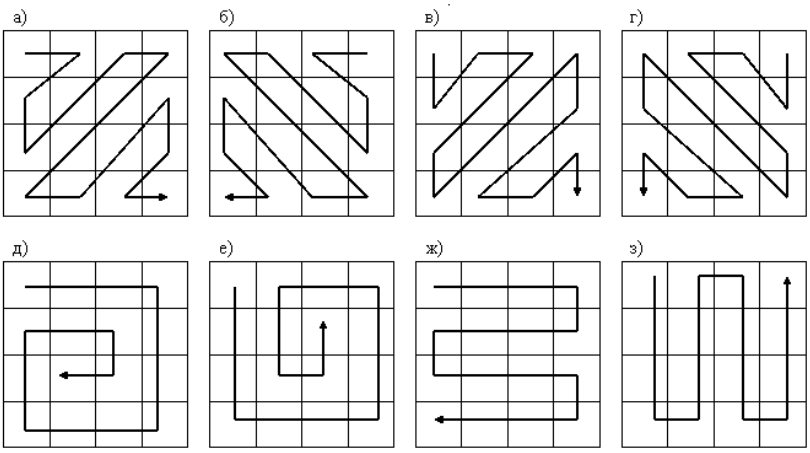

Рис. 1: Порядок расположения элементов матрицы

ствует. Обратную матрицу можно найти с помощью следующего метода: если линейными преобразованями строк привести исходную матрицу к единичной, то этими же преобразованиями единичная матрица приводится к искомой обратной матрице.

6.30. Заполнение матрицы по змейке. Дано натуральное $N (N \leqslant 10)$. Заполнить квадратную матрицу размером $N \times N$ целыми числами $0, 1, 2, \ldots, N^2 - 1$ в соответствии со схемой расположения (см. рис. 1).

6.31. Сортировка элементов матрицы. Даны натуральное $N (N \leqslant 10)$, значения элементов целочисленной квадратной матрицы размером $N \times N$. Отсортировать матрицу таким образом, чтобы при обходе её элементов в порядке, представленном на рис. 1, они бы располагались по неубыванию. Использовать метод:

а) сортировки выбором (см. задачу 5.27);

б) сортировки вставками (см. задачу 5.28);

в) пузырьковой сортировки (см. задачу 5.29);

г) сортировки подсчетом (см. задачу 5.30);

6.32. Сортировка матрицы с ограничением. Решить задачу 6.31 в условиях следующего ограничения: не допускается использовать дополнительные структуры данных (например, линейные массивы), то есть сортировку нужно делать «на месте» — непосредственно в исходной матрице. Размер матрицы N вводится с клавиатуры.

Замечания: если для вычисления координат каждого элемента матрицы напрямую использовать циклические алгоритмы, как в задаче 6.30, то это существенно увеличивает временную сложность исходного алгоритма (по крайней мере, на два порядка). Рекомендуется найти такой алгоритм, в котором время выполнения сортировки было бы пропорционально квадрату количества сортируемых элементов (то есть так, как в случае с линейным массивом). Для сортировки методом подсчета ограничение ослабляется — разрешается использовать дополнительную матрицу счетчиков.

6.33. Решить задачу 6.32 для прямоугольной матрицы размером $M \times N$, где размеры вводятся, например, с клавиатуры.

7. Строки

В задачах этого раздела используется строковый тип данных (строка символов). Решение некоторых задач может зависеть от используемой в операционной системе кодировки русского алфавита. Предлагается указанную проблему читателям решить самостоятельно.

А

7.1. Дана символьная строка s, символ c. Найти количество вхождений символа c в строку s.

7.2. Дана символьная строка s, символы c_1, c_2. Заменить все вхождения символа c_1 в строку s на c_2.

7.3. Даны две символьных строки s_1, s_2. Найти количество вхождений строки s_2 в строку s_1.

7.4. Даны три символьных строки s_1, s_2, s_3. Заменить все вхождения строки s_2 в исходную строку s_1 на строку s_3.

7.5. Даны две символьных строки s_1, s_2. Удалить все вхождения строки s_2 в исходную строку s_1.

7.6. Даны две символьных строки s_1, s_2. Проверить, все ли символы строки s_2 входят в строку s_1.

7.7. Все прописные буквы данного русского текста заменить на строчные, и наоборот.

7.8. Дана строка. Выяснить, является ли эта строка идентификатором, десятичной или двоичной записью целого числа. *Идентификатор* – это последовательность латинских букв, цифр, начинающаяся с буквы.

7.9. Дано натуральное число n. Получить строковое представление последовательности цифр числа n, отделяя друг от друга пробелами группы по три цифры, начиная справа.

7.10. Дана строка букв русского алфавита. Найти количество гласных и количество согласных букв.

7.11. Дана строка букв русского алфавита. Определить, упорядочены ли буквы по алфавиту.

7.12. Дана строка букв русского алфавита. Упорядочить буквы по алфавиту.

7.13. Считается, что в тексте соблюден баланс круглых скобок, если каждой открывающей круглой скобке далее по тексту соответствует закрывающая скобка. Определить, имеется ли баланс круглых скобок в данной строке.

B

7.14. Дано натуральное число n. Присвоить строковой переменной s это число, записанное русскими словами (например, для $n = 13951$ значение s должно получиться равным "тринадцать тысяч девятьсот пятьдесят три").

7.15. Решить задачу 7.14 для вещественного числа r, представленного в десятичной записи.

7.16. В тексте могут быть скобки трех видов: круглые ( ), квадратные [ ], фигурные { }. Считается, что в тексте соблюден баланс скобок, если открывающей скобке каждого вида далее по тексту соответствует закрывающая скобка того же вида, причем часть текста между этими скобками также сбалансирована. Дана строка. Определить, соблюден ли баланс трех видов скобок в этой строке.

7.17. Дан текст, сбалансированный по круглым скобкам. Заменить все символы между скобками (кроме других скобок и их нового содержимого) на символ "*". Например, из строки "(ab(cd)e(fd))" должна получиться строка "(*(*)(*))".

В задачах 7.18–7.35 *словом* в строке будем считать последовательность символов русского алфавита; слова отделяются друг от друга пробелами или знаками препинания. Регистр букв, если это не указано особо, не различать, то есть слова "Слово" и "сЛоВо" считать одинаковыми. Текст – это строка символов.

7.18. В данном тексте найти самое длинное слово.

7.19. В данном тексте найти все слова-палиндромы, то есть такие слова, которые одинаково читаются слева направо и справа налево.

7.20. В данном тексте найти слово, содержащее наибольшее количество различных букв.

7.21. В данном тексте первые буквы слов сделать прописными, а остальные буквы – строчными.

7.22. В данном тексте найти все слова, буквы в которых упорядочены по алфавиту.

7.23. В данном тексте найти все такие слова, в которых все буквы различны.

7.24. В данном тексте найти все слова, в которых гласные буквы чередуются с согласными.

7.25. В данном тексте в каждом слове переставить буквы так, чтобы они были упорядочены по алфавиту.

7.26. В каждом слове данного текста оставить только первое вхождение каждой буквы в это слово.

7.27. В данном тексте найти наиболее часто встречающиеся слова.

7.28. В данном тексте переставить местами слова так, чтобы они располагались в алфавитном порядке. Чередование знаков препинания при этом нужно сохранить. Например, текст "читайте, завидуйте: я – гражданин!" должен быть преобразован в "гражданин, завидуйте: читайте – я!".

7.29. Преобразовать входной текст, как в задаче 7.28, только слова разместить в обратном порядке относительно исходного.

7.30. Из данного текста удалить все слова, входящие в него в точности n раз.

7.31. Из данного текста удалить повторные вхождения слов.

7.32. Назовем *предложением-палиндромом* такой текст, в котором первое слово совпадает с последним, второе – с предпоследним и т.д. Проверить, является ли входной текст предложением-палиндромом.

7.33. Назовем *фразой-палиндромом* текст, который по буквам читается одинаково как с начала, так и с конца. Проверить, является ли данный текст фразой-палиндромом. Например, фразы-палиндромы: "А роза упала на лапу Азора", "Аргентина манит негра"и "Он дивен – палиндром, и ни морд, ни лап не видно".

7.34. Выравнивание. Дана строка s, натуральное $n \leqslant 20$. Разбить строку s на последовательность подстрок так, чтобы в каждой подстроке находились целые слова (если это возможно) и чтобы длина каждой подстроки не превышала n. Количество подстрок должно быть минимальным. Если длина слова больше n, то разрешается разбить это слово. Вывести на экран подстроки следующими способами:

а) с выравниванием по левому краю;
б) с выравниванием по правому краю;
в) с выравниванием по центру;
г) с выравниванием по ширине, то есть за счет равномерного увеличения количества пробелов между словами добиться того, чтобы длина каждой подстроки (кроме, возможно, последней) равнялась бы n.

7.35. Под *переворотом согласных* назовем такую перестановку букв в слове, что первая согласная меняется местами с последней согласной, вторая – с предпоследней и т.д. Произвести переворот согласных во всех словах входного текста, имеющих наибольшую длину.

7.36. Дана строка из двух символов, первый – латинская буква от "a"до "h", второй – цифра от 1 до 8. Рассматривая символы как координаты ферзя на шахматной доске, нарисовать на экране доску; клетки, которые бьет ферзь, отметить крестиками, все остальные – точками.

C

7.37. Дано натуральное число n в десятичной записи. Найти запись этого числа в системе счисления с основанием R ($2 \leqslant R \leqslant 36$).

7.38. Дано вещественное число a в десятичной записи. Найти запись этого числа в системе счисления с основанием R ($2 \leqslant R \leqslant 36$) (возможно появление периода).

7.39. Дано натуральное число n, записанное в системе счисления с основанием R ($2 \leqslant R \leqslant 36$). Найти запись этого числа в десятичной системе счисления.

7.40. Дано вещественное число a, записанное в системе счисления с основанием R ($2 \leqslant R \leqslant 36$). Найти запись этого числа в десятичной системе счисления (возможно появление периода).

7.41. Дано натуральное число n, записанное арабскими цифрами. Найти запись этого числа римскими цифрами (римские цифры означают: $I-1, V-5, X-10, L-50, C-100, D-500, M-1000$).

7.42. Дано число n, корректно записанное римскими цифрами. Найти десятичную запись этого числа арабскими цифрами.

7.43. Дано натуральное $R, 2 \leqslant R \leqslant 16$. Напечатать таблицу умножения в системе счисления с основанием R.

7.44. Преобразовать текст, написанный русскими буквами (кириллицей), в текст, написанный соответствующими латинскими буквами (латиницей). Например, текст "Читайте, завидуйте: я – гражданин!" должен быть преобразован в текст "CHitaite, zaviduite: ya – grazhdanin!".

7.45. Преобразовать текст, написанный латиницей, в текст, написанный кириллицей (см. задачу 7.44).

7.46. Дан текст. Найти слова текста, которые можно составить из других слов этого же текста. Например, в тексте "бур ура кобура бура ко ра кобур"слово "кобура"можно составить из слов "ко"и "бура".

7.47. Дан русский текст. Найти в тексте все пары слов, каждое из которых можно получить переворотом второго слова; например, слова ("клоп", "полк") образуют такую пару.

7.48. Дан русский текст. Найти в тексте все пары слов, каждое из которых можно получить перестановкой букв второго слова; например, слова ("слово", "волос") образуют такую пару.

7.49. Даны две строки s_1, s_2. Найти самую длинную общую подстроку этих строк.

7.50. Даны две строки s, w. Строка s представляет собой шаблон слова, то есть может состоять из русских букв и знаков "*" и "?". Знак "*" означает любую последовательность символов, в том числе пустую; знак "?" означает любой единичный символ. Строка w – слово из русских букв. Проверить, соответствует ли слово w шаблону s. Например, шаблону "ко*о*?" слово "компромат" соответствует, а слово "колесо" не соответствует.

7.51. Дан русский текст. Проверить правописание гласных в слогах *жи-ши-, ча-ща-, чу-щу-, ци-* с учетом исключений (*парашют, жюри, брошюра, цыган* и др.). В случае обнаружения ошибки – исправить.

7.52. Дан текст на одном из популярных языков программирования (Basic, Pascal, C, Prolog и др.). Определить язык программирования, на котором написан текст.

7.53. Алгоритм Рабина. Решить задачу поиска подстроки в строке. При прямом побуквенном сравнении задача решается долго. Предлагается другой алгоритм: допустим, есть подстрока $w_1 w_2 \ldots w_n$, которую нужно обнаружить в строке

$s_1s_2\dots s_m (n \leqslant m)$. Вычислим некоторую однозначную функцию на строках $w_1w_1\dots w_n$ и $s_1s_2\dots s_n$. Если значения функции не совпали, то эти строки не совпадают и нужно перейти к следующему шагу. На следующем шаге сравним значения функции на строках $w_1w_1\dots w_n$ и $s_2s_3\dots s_{n+1}$ и так далее до исчерпания строки $s_1s_2\dots s_m$; если на каком-то шаге значения функции совпали, то только тогда имеет смысл проводить побуквенное сравнение. Возникает вопрос, а чем лучше вычисление функции на строке побуквенного её сравнения с другой строкой. Ответ прост — каждая следующая за подстрокой $s_is_{i+1}\dots s_{i+n-1}$ подстрока может быть получена из предыдущей только удалением первого символа s_i и добавлением нового s_{i+n} . Поэтому, если подобрать «хорошую» функцию, которую не нужно заново полностью перевычислять, а можно воспользоваться вычисленным на предыдущем шаге значением функции, то действительно алгоритм дает некоторое преимущество по сравнению с прямым перебором. Примером такой функции может служить сумма кодов входящих в строку символов, тогда вычисление нового её значения будет заключаться в вычитании кода первого символа и прибавлением кода нового символа. Реализовать предложенный алгоритм. Попробуйте найти другие функции и оцените выигрыш по времени.

7.54. Даны две строки симоволов s_1 и s_2. Найти наиболее длинную общую подстроку s_1 и s_2, то есть такую строку s_3, чтобы она входила как подстрока в каждую из строк s_1 и s_2 и имела наибольшую длину из всех таких строк.

7.55. Вхождение по шаблону. Дан шаблон и строка символов (как в задаче 7.50). Найти в строке наидлиннейшую подстроку, подходящую под данный шаблон, или сообщить, что таковой не существует.

Пример.

Шаблон:*a*?b?c*

Строка:*bbaabacbccc*

Ответ: Подходит подстрока = aabacbc

7.56. Без комментариев. Дана программа на языке Паскаль. В тексте могут быть комментарии, ограниченные фигурными скобками, либо скобками вида (* и *). Вывести текст программы без комментариев.

7.57. Период строки. Периодом строки $s = a_1a_2\dots a_n$ назовем такое целое положительное число p, что строка s может быть представлена как сцепление (конкатенация) k одинаковых строк длиной $p (k \geqslant 1)$, то есть $s = (w)^k = \underbrace{ww\dots w}_{k}$, где $w = b_1b_2\dots b_p$.

Дана строка s. Найти все её периоды.

Пример.

Вход: *abababab*

Выход: Периоды строки:

8 (w=abababab, k=1)

4 (w=abab, k=2)

2 (w=ab, k=4)

7.58. Строки Фибоначчи. Определим последовательность строк Фибоначчи $s_1, s_2, \dots, s_k, \dots$ следующим образом: $s_1 =$ $"b", s_2 = "a", s_k = s_{k-1}s_{k-2} (k > 2)$. Тогда начало последовательности строк Фибоначчи выглядит так:

b a ab aba abaab abaababa abaababaabaab ...

Решить следующие задачи:

а) для данного k найти длину s_k;

б) для данного k найти s_k;

в) дана строка s; проверить, не является ли она строкой Фибоначчи?

7.59. Обобщенный палиндром. Введем рекуррентное определение: строка символов s называется *обобщенным палиндромом* (ОП), если выполняется в точности одно из трех условий:

а) s — пустая строка (то есть не имеет ни одного символа);

б) s состоит из одного символа;

в) s можно представить так: $s = uxu$, где u— непустая строка, а x является ОП (то есть префикс и постфикс совпадают, а между ними — ОП).

Например, строки "abcdedabc", "abaababaabaab" являются ОП. Дана строка. Определить, является ли она обобщенным палиндромом?

7.60. Расстояние Хемминга. Расстоянием Хемминга между строками одинаковой длины s_1 и s_2 называется количество несовпадающих симоволов в s_1 и s_2, стоящих на одинаковых позициях. Даны две строки. Найти расстояние Хемминга между ними.

7.61. Расстояние. Расстоянием между строками s_1 и s_2 называется минимальное количество вставок и удалений символов из строки s_1 для преобразования её в строку s_2. Даны две строки. Найти расстояние между ними.

7.62. Расстояние редактирования. Расстоянием редактирования между строками s_1 и s_2 называется минимальное количество вставок, удалений и замен символов из строки s_1 для преобразования её в строку s_2. Даны две строки. Найти расстояние редактирования между ними.

7.63. Подсказка при вводе. Программа вводит последовательность слов, каждое — с новой строки. В процессе ввода каждого слова программа предлагает возможные варианты продолжения, если введенная начальная часть слова совпадает с началом одного или нескольких ранее введенных слов. Например, после ввода слов "компилятор" и "коммунизм", при вводе слова первых трех букв слова "компьютер" программа должна предоставить список возможных продолжений ("пилятор", "мунизм"). Пользователь должен иметь возможность выбора одного из предлагаемых вариантов или продолжить ввод слова. После ввода пустой строки программа должна выдать список всех введенных слов (без повторов) и завершить работу. Вопрос для обдумывания: какая структура хранения слов будет наиболее оптимальной с точки зрения времени поиска и исключения повторных просмотров массива слов?

7.64. Метаграмма «Из мухи слона». Существует интересная словесная игра (метаграмма). Требуется из слова s_1 получить слово s_2 за конечное число преобразований. Игра начинается со слова s_1, его заменяют таким словом, чтобы оно отличалось от исходного не более чем на одну букву. Затем заменяют новое слово и так далее до тех пор, пока не получим слово s_2. Цепочка полученных слов (если она существует) является решением метаграммы. Дан словарь слов (например, русских), и даны два слова s_1 и s_2 одинаковой длины. Построить цепочку преобразований из слова s_1 в слово s_2.

Пример.

(Считается, что словарь слов задан.)

Вход: *муха слон*

Выход: муха - мура - тура - тара - кара - каре - кафе - кафр - каюр - каюк - крюк - урюк - урок - срок - сток - стон - слон

7.65. Метаграмма М. Гервера. Немного изменим правила игры в метаграммы (см. задачу 7.64). За каждый ход в слове не только заменяется одна буква на другую, но также разрешается менять порядок букв в слове. Задание то же: построить цепочку преобразований из исходного слова в конечное.

Пример.

(считается, что словарь слов задан)

Вход:*муха слон*

Выход: муха - хула - луна - лунь - ноль - слон

7.66. Нарезать строку. Дана строка символов. Найти минимальное число разрезов, чтобы получившиеся части представляли собой неубывающие или невозрастающие последовательности символов (символы сравнивать в соответствии с их кодами, например ASCII).

8. Множества

В задачах этого раздела рекомендуется применять множественный тип языка программирования. Для тех языков, в которых отсутствует встроенный тип-множество, можно его самостоятельно реализовать (например, в виде библиотечного модуля или класса). К сожалению, даже в тех языках, где имеется множественный тип, он обладает недостатками, например существенной ограниченностью мощности (количества элементов) множества. Поэтому иногда в задачах этого раздела мы вынуждены указывать довольно искусственное ограничение количества элементов – 256. Читателю рекомендуется попытаться это ограничение преодолеть.

A

8.1. Дано натуральное число n. Найти количество различных значащих цифр десятичной записи числа n.

8.2. Дано натуральное число n. Найти цифры, не входящие в десятичную запись числа n.

8.3. Дана строка символов s. Найти множество символов, входящих в строку s.

8.4. Даны строки s_1 и s_2. Найти все символы, входящие в обе эти строки одновременно.

8.5. Даны строки s_1 и s_2. Найти все символы, входящие в s_1, но не входящие в s_2.

8.6. Даны строки s_1 и s_2. Найти все символы, входящие хотя бы в одну из этих строк.

8.7. Дан линейный массив натуральных чисел $a_i, a_i \geqslant 255, i = 1, 2, \ldots N$. Повторные вхождения элементов массива заменить на нули.

8.8. Дана строка символов. Найти множество латинских символов, отсутствующих в строке.

8.9. Дан русский текст. Найти отношение количества гласных букв к количеству согласных букв в тексте.

8.10. В возрастающем порядке напечатать натуральные числа, не превосходящие 255, которые можно представить в виде $n^2 + k^2; n, k$ – натуральные числа.

B

8.11. Дано натуральное число n. Найти основание системы счисления R $(2 \leqslant R \leqslant 36)$, в которой число n записывается с помощью наименьшего количества различных цифр.

8.12. Даны натуральные числа n, R $(2 \leqslant R \leqslant 36)$. Найти цифры, не входящие в запись числа n в системе счисления с основанием R.

8.13. Дано натуральное число $n, 1 \leqslant n \leqslant 255$. Найти наибольшее из натуральных чисел k, не превосходящее n, которое представимо в виде $2^x 3^y 5^z$, где x, y, z – целые неотрицательные числа.

8.14. Дана строка, состоящая из русских слов (определение слова см. в пояснении к задаче 7.18). Найти:

а) все гласные буквы, входящие в каждое слово;
б) все согласные буквы, не входящие ни в одно слово;
в) все звонкие согласные буквы, которые входят в более чем одно слово;
г) все глухие согласные буквы, которые входят хотя бы в одно слово неоднократнно;
д) все гласные буквы, входящие ровно в одно слово;
е) все согласные буквы, входящие в более чем одно слово.

C

В задачах 8.15–8.18 рекомендуется использовать *метод решета.* Суть метода заключается в том, что для получения некоторого множества элементов, обладающих определенным свойством, вначале строят множество всех возможных элементов, а затем из него последовательно исключают элементы, не обладающие указанным свойством.

8.15. Решето Эратосфена. Построить множество всех простых чисел, не превосходящих N. Для этого необходимо вначале построить множество всех натуральных чисел от 2 до N. Затем выбирают наименьшее из непросмотренных чисел (то есть 2) и исключают из множества все числа, кратные 2, кроме самого этого числа. Затем повторяют тот же шаг для $3, 5, \ldots$, пока в множестве имеются непросмотренные числа. В результате в множестве останутся только простые числа.

8.16. Найти все простые и все составные числа Фибоначчи, не превосходящие 255 (см. задачи 8.15, 3.20).

8.17. Средневековая задача о старушке и яйцах. Старушка шла с корзиной яиц на рынок, когда её сбил с ног всадник, и все яйца разбились. Всадник предложил возместить убытки и спросил старушку, сколько у неё было яиц в корзине. Старушка ответила, что точно не помнит, но когда она брала яйца по два, оставалось одно яйцо, когда она брала по 3, 4, 5, 6 яиц в остатке было одно яйцо. Но когда она брала по 7 яиц, остатка не было. Сколько могло быть яиц в корзине старушки? Задачу обобщить, приняв возможным различные значения делителей и остатков. Предупреждение: стандартного множественного типа может не хватить!

8.18. Счастливые числа. Имеем последовательность натуральных чисел $1, 2, 3, 4, 5, 6, 7 \ldots$. Число 1 оставляем в последовательности. Число 2 считаем «просеивающим» и исключаем каждое второе число последовательности, получается последовательность $1, 3, 5, 7, \dots$. Следующее «просеивающее» число – 3, поэтому исключаем каждое третье число, получаем последовательность $1, 3, 7, 9, 13, \dots$. Затем исключаем каждое седьмое число, каждое девятое и так далее. Числа, которые никогда не будут исключены из последовательности, называются счастливыми. Получить все счастливые числа, не превосходящие 255.

8.19. Проверить следующие тождества для произвольных значений множеств целых чисел A, B, C:

а) $A \cup (B \cap C) = (A \cup B) \cap (A \cup C)$;
б) $(A \setminus B) \setminus C = A \setminus (B \setminus C)$;
в) $(A \setminus B) \cup (A \setminus C) = A \setminus (B \setminus C)$;
г) $(A \setminus B) \cup (B \setminus C) = (A \setminus C) \cup (C \setminus B)$;
д) $(A \bigtriangleup B) \bigtriangleup (C \bigtriangleup D) = (A \bigtriangleup D) \bigtriangleup (C \bigtriangleup B)$[2];
е) $(A \bigtriangleup (B \bigtriangleup C)) = (A \cup B \cup C) \setminus (A \cap B) \setminus (A \cap C) \setminus (B \cap C)$;
ж) $(A \setminus B) \cup (B \setminus A) = (A \cup B) \setminus (B \cap A)$.

9. Записи (структуры)

В задачах этого раздела необходимо использовать структурный тип языка программирования – запись (структура). Как правило, в виде структуры реализуются некоторые составные понятия, разлагающиеся на отдельные составляющие – поля. Исходные данные для программ удобнее всего хранить в текстовых файлах.

А

9.1. Каждый вид товара имеет наименование, цену (в рублях

[2] Определение симметрической разности множеств см. в примечании к задаче 5.23.

и копейках) и срок гарантии. Выдать на экран список товаров по невозрастанию цены; по неубыванию срока гарантии; в алфавитном порядке наименований.

9.2. Точки на плоскости заданы именами (A, B, C и т.д.) и декартовыми координатами. Для всех точек вычислить и выдать их полярные координаты; выдать список точек по неубыванию расстояния от точки до центра координат; найти площадь треугольника по трем точкам, заданным своими именами.

9.3. Даны несколько рациональных чисел с выделенной целой частью и дробной частью в виде натуральных числителя и знаменателя. Найти произведение всех чисел в виде рационального числа такого же вида с несократимой дробной частью; сумму всех чисел; наименьшее и наибольшее числа; среднее арифметическое всех чисел.

9.4. Информация о каждом студенте состоит из фамилии, имени, отчества, номера группы, трех оценок за сессию. Выдать список студентов в алфавитном порядке; по неубыванию средних оценок.

9.5. Каждая игральная карта характеризуется мастью (пики, трефы, бубны, червы) и достоинством (от шестерки до туза). Даны несколько игральных карт. Выдать список карт в порядке увеличения старшинства, считая, что масти упорядочены так: пики, трефы, бубны, червы. Проверить, бьет ли первая карта исходного списка вторую с учетом того, что M – козырная масть.

9.6. Дано несколько комплексных чисел вида $z = a + bi$. Каждое число представить в тригонометрической форме $z = r(\cos\varphi + i\sin\varphi)$ и выдать на экран список чисел в порядке неубывания модулей; в порядке невозрастания углов φ; в порядке невозрастания мнимых частей b.

9.7. Дан список сотрудников организации, о каждом известны его фамилия, имя, отчество, дата рождения. Дополнить информацию о знаках зодиака для каждого сотрудника. Выдать на экран список сотрудников в алфавитном порядке; в порядке неубывания дат рождений; в алфавитном порядке знаков зодиака.

9.8. В автопарке ведется учет движения городских автобусов. По каждому автобусу известны его госномер, номер маршрута, фамилия водителя, время выхода из парка, время возвращения в парк. Выдать на экран список автобусов в порядке неубывания номеров маршрутов; в алфавитном порядке фамилий водителей; в порядке неубывания времени нахождения на маршруте.

9.9. Библиографический каталог содержит информацию о книгах библиотеки, в карточке каждой книги укзаны название, автор, издательство, год выпуска, фамилии трех последних читателей, читавших книгу. Выдать на экран список книг в алфавитном порядке названий; по неубыванию годов выпуска; список книг, которые читал в последнее время читатель с указанной фамилией.

9.10. Наружное наблюдение спецслужбы за объектом фиксирует все происходящие с объектом события по их наименованиям и время возникновения (год, месяц, число, час, минута). Выдать список событий в хронологическом поряке; найти два самых далеких по времени события, два самых близких события.

9.11. Дана расстановка шахматных фигур на доске. Каждая шахматная фигура характеризуется названием, цветом, позицией (буква от a до h и цифра от 1 до 8). Выдать список белых и список черных фигур по неубыванию позиций (считать, что позиции упорядочены в первую очередь по буквам, а затем – по цифрам); указать, какие фигуры находятся под боем ферзя соперника.

9.12. В профкоме организации ведется учет сотрудников, стоящих в очереди на получение жилья. По каждому сотруднику известны его фамилия, имя, отчество, стаж в годах, табельный номер, табельный номер другого сотрудника, который находится непосредственно перед ним в очереди на жильё. Табельный номер впередистоящего перед первым в очереди сотрудником считать равным (-1). Выдать список сотрудников в алфавитном порядке; в порядке очередности на получение жилья.

9.13. В записной книжке хранятся фамилии, имена, отчества, адреса и номера телефонов людей. Выдать на экран список людей в алфавитном порядке; в алфавитном порядке адресов; по неубыванию номеров телефонов; номера телефонов всех людей с указанной фамилией.

9.14. В расписании движения поездов по станции хранится информация о каждом проходящем поезде – номер поезда, начальный и конечный пункты движения, время прибытия, стоянки и отправления. Выдать на экран список поездов по неубыванию времени отправления; по невозрастанию номеров поездов; всех поездов с заданным конечным пунктом.

9.15. Р асписание движения самолетов хранит информацию о номере рейса, пункте назначения, дате и времени вылета, типе самолета. Выдать на экран список рейсов в порядке неубывания даты и времени вылета; список различных типов самолетов; список рейсов, направляющихся в указанный пункт назначения.

9.16. В банке о каждом расчетном счёте клиента хранится следующая информация: номер счёта, фамилия клиента, ИНН клиента, пять последних операций по счёту (дата операции, вид операции (приход или расход), сумма). Выдать на экран список счетов в алфавитном порядке фамилий клиентов; хронологический список проведенных операций по счетам; по неубыванию сумм остатков на счетах, считая, что до выполнения первой из операций на каждом счете средств не было.

B

9.17. Даны комплексные коэффициенты многочлена $P(x) = z_n x^n + z_{n-1}x^{n-1} + \cdots + z_1 x^1 + z_0$. Найти значение многочлена $P(x)$ в точке x, где $x = a + bi$ – комплексное число.

C

9.18. Машина Тьюринга. Смоделировать работу машины Тьюринга по заданной входной ленте $a_{i_0} a_{i_1} \ldots a_{i_k}$, функции перехода, состоящей из нескольких правил, каждое из которых имеет вид $q_l a_m \to a_n q_p M$, где $M \in \{L, R, S\}$, начальному состоянию q_0 и множеству конечных состояний $\{q_{f_1}, q_{f_2}, \ldots, q_{f_r}\}$.

9.19. Нормальный алгоритм Маркова. Смоделировать выполнение нормального алгоритма А.А. Маркова по заданной входной строке s и последовательности правил, каждое из которых имеет вид $s_i \to s_j$ или $s_i \mapsto s_j$ (s_i и s_j - строки).

10. Подпрограммы

При решении задач данного раздела необходимо использовать механизм подпрограмм, то есть часть текста программы, решающую локальную задачу, оформить в виде отдельной функции или процедуры с необходимым набором входных или выходных параметров.

A

10.1. По данным значениям аргументов a и b вычислить значение функции $f(a,b)$. При организации вычисления необходимо выявить сходные по своей схеме последовательности операций, различающиеся лишь операндами, и представить их в виде отдельных функций. Аргументы должны принадлежать области определения функции:

а) $f(a,b) = \max^2(a,b) + \max^2(a-b, a^2-b^2) - \max^4(\frac{a-b}{2}, \frac{a+b}{2})$;

б) $f(a,b) = \dfrac{1+a+a^2+a^3+a^4}{1+\frac{1}{b}+\frac{1}{b^2}+\frac{1}{b^3}+\frac{1}{b^4}}$;

в) $f(a,b) = \dfrac{\max(a+b, a-b)}{\max(b+a, b-a)} + \dfrac{\max(a^2+b^2, a^2-b^2)}{\max(b^2+a^2, b^2-a^2)}$;

г) $f(a,b) = \dfrac{a+b+\sqrt{a^2+ab+b^2}}{-\frac{a}{b}+\frac{a^2}{b^2}-\frac{a^3}{b^3}+\frac{a^4}{b^4}} + \dfrac{\frac{a}{b}-\frac{b}{a}+\sqrt{\frac{a^2}{b^2}-1+\frac{b^2}{a^2}}}{a+a^2+a^3+a^4}$;

д) $f(a,b) = a^a + b^{a+1} + b^b + a^{b+1} + (a+b)^{a+b} + b^{a+b+1}$;

е) $f(a,b) = \dfrac{\sqrt{a}+\sqrt{b}}{(\frac{1}{a})^2+(\frac{1}{b})^2} - \dfrac{\frac{1}{\sqrt{a}}+\frac{1}{\sqrt{b}}}{a^2+b^2}$;

ж) $f(a,b) = \dfrac{\frac{1}{a}+\frac{1}{b}+\frac{1}{a^2}+\frac{1}{b^2}+\frac{1}{ab}}{(a+b)+(a-b)+(a+b)^2+(a-b)^2+(a^2-b^2)}$;

з) $f(a,b) = \ln\left(\frac{1}{a^2}+\frac{1}{b^2}\right) + \ln^2(a^2+b^2) + \ln^3((a+1)^2+(b+1)^2)$;

и) $f(a,b) = (a^2+ab+b^2)^2 + (a^2+ab+b^2)b + b^2$;

к) $f(a,b) = \dfrac{\frac{a+b}{a-b}+a}{\frac{a+b}{a-b}-a}$;

л) $f(a,b) = \sqrt{\sqrt{a+1+\frac{1}{b}}+\frac{1}{b+1}}$;

м) $f(a,b) = (2+\frac{1}{2}+a+\frac{1}{a}) + \dfrac{1}{2+\frac{1}{2}+a+\frac{1}{a}} + b + \frac{1}{b}$;

н) $f(a,b) = \dfrac{\dfrac{ab}{a+b}\cdot b}{\dfrac{ab}{a+b}+b}$;

о) $f(a,b) = \dfrac{\sqrt{a+b}+\sqrt[3]{a-b}+\sqrt[4]{a+b}+\sqrt[5]{a-b}}{\sqrt{ab}+\sqrt[3]{\dfrac{a}{b}}+\sqrt[4]{ab}+\sqrt[5]{\dfrac{a}{b}}}$;

п) $f(a,b) = \dfrac{\dfrac{a^2-b^2}{a^2+b^2}-a-b}{\dfrac{a^2-b^2}{a^2+b^2}+a+b}$;

р) $f(a,b) = (a^2+b^2-ab)^2+(ab)^2-(a^2+b^2-ab)(ab)$.

10.2. Даны четыре положительных вещественных числа a, b, c, d. Для каждой тройки этих чисел определить, существует ли треугольник с такими сторонами и среди всех треугольников найти тот, у которого площадь максимальна.

10.3. Даны длины сторон некоторого треугольника a, b, c. Найти величины углов этого треугольника в градусах и длины его медиан.

10.4. Даны натуральные a, b и c. Найти наибольший общий делитель и наименьшее общее кратное этих трех чисел.

10.5. Решить задачу 10.4 для четырех натуральных чисел a, b, c, d.

10.6. Два простых числа называются *близнецами*, если они отличаются друг от друга на 2. Найти все пары близнецов из первых двух сотен натуральных чисел.

10.7. Два натуральных числа называются *дружественными*, если каждое из них равно сумме всех делителей другого числа, кроме него самого, например, числа 1184 и 1210 – дружественные. Найти все пары дружественных чисел из первых пяти тысяч натуральных.

10.8. Даны вещественные $a_1, a_2, b_1, b_2, c_1, c_2$. Если все корни одного из уравнений $a_1x^2 + b_1x + c_1 = 0, a_2x^2 + b_2x + c_2 = 0$ больше корней другого уравнения, то выдать на экран 1, во всех остальных случаях выдать 0.

10.9. Даны вещественные a, b, c. Среди всех перестановок этих чисел (a, b, c), (a, c, b), (b, a, c), $\ldots$ найти такую, чтобы квадратное уравнение, коэффициенты которого совпадают с соответствующими числами в перестановке, имело бы два вещественных корня и расстояние на числовой оси между корнями было бы наибольшим. Если такой перестановки не существует, сообщить об этом.

10.10. Даны вещественные n-мерные векторы x, y, z. Найти:
- а) $xy + yz + xz$;
- б) $x^2 + y^2 + z^2$;
- в) $|x| + |y| + |z|$;
- г) $2x + 3y + 4z$;
- д) $(x + y)(x + z) + (x + y)(y + z) + (x + z)(y + z)$.

10.11. Даны вещественные квадратные матрицы A, B порядка n, n-мерные вещественные векторы x, y. Найти:
- а) $Ax - Ay - Bx + By$;
- б) $Ax \cdot Ay - Bx \cdot By$;
- в) $(A + B)x \cdot (A - B)y - \dfrac{Ax \cdot By}{Bx \cdot Ay}$.

10.12. Даны вещественная квадратная матрица A порядка n, n-мерный вещественный вектор x. Среди векторов $x, Ax, Ax - x$ найти тот, который имеет наибольшую длину.

10.13. Даны вещественные квадратные матрицы A, B порядка n. Найти:
- а) $AB + BA$;
- б) $A^2 + B^2 + (AB)^2$;
- в) $(A + B)^2 + (A - B)^2$.

В

10.14. Даны вещественные квадратные матрицы A, B порядка

n. Найти:

а) $\det A + \det B - \det(A+B)$;

б) $\det(A^2) + \det(B^2) - \det(AB)$;

в) $\det(A+B) + \det(AB) - \det(A^2+B^2)$;

г) $\max(\det A, \det B, \det(AB))$.

10.15. Привести выражение к несократимой обыкновенной дроби:

а) $1 + \frac{1}{2} + \frac{1}{3} + \cdots + \frac{1}{10}$;

б) $\frac{1}{2} + \frac{2}{3} + \cdots + \frac{9}{10}$;

в) $\frac{1}{2} + \frac{1}{1\cdot 2} + \frac{1}{1\cdot 2\cdot 3} + \frac{1}{1\cdot 2\cdot 3\cdot 4} + \frac{1}{1\cdot 2\cdot 3\cdot 4\cdot 5}$.

10.16. Дан массив натуральных чисел $a_1, a_2, \ldots, a_n$. Найти наибольший общий делитель и наименьшее общее кратное всех чисел массива.

10.17. Даны два массива натуральных чисел $a_1, a_2, \ldots, a_n$ и $b_1, b_2, \ldots, b_n$. Найти несократимую обыкновенную дробь, равную

а) $\frac{a_1}{b_1} + \frac{a_2}{b_2} + \cdots + \frac{a_n}{b_n}$;

б) $\frac{a_1}{b_1} - \frac{a_2}{b_2} + \cdots \pm \frac{a_n}{b_n}$;

в) $\frac{a_1}{b_1+b_2} + \frac{a_2}{b_2+b_3} + \cdots + \frac{a_n}{b_n+b_1}$.

10.18. Даны вещественные a_1, a_2, $\ldots$, a_n;b_1, b_2, $\ldots$, b_n;c_1, c_2, $\ldots$,c_n $(n \geqslant 3)$. Среди прямых $a_i x + b_i y + c_i = 0$ найти три таких, которые при пересечении образуют треугольник наибольшей площади или сообщить, что таких прямых не существует.

10.19. Даны координаты различных точек плоскости (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n)$, где $n \geqslant 3$. Найти тройку точек, которые являются вершинами треугольника наибольшей площади.

C

10.20. Параметры-функции. По данному значению натурального аргумента n вычислить значение функции $f(n)$. При организации вычисления необходимо выявить сходные по своей схеме последовательности операций, различающиеся лишь операндами и/или используемыми функциями, и представить их в виде отдельных функций. Рекомендуется в качестве параметров передавать функцию (процедурный тип). Аргументы должны принадлежать области определения функции:

а) $f(n) = \dfrac{1^2 + 2^2 + \cdots + n^2}{cos(1) + cos(2) + \cdots + cos(n)}(ln(1) + ln(2) + \cdots + +ln(n))$;

б) $f(n) = max\{\sum_{i=1}^{n} sin^i i, \sum_{i=1}^{n} cos^i i, \sum_{i=1}^{n} ctg^i i\}$.

10.21. Реализовать арифметические операции с *длинными числами*, то есть с целыми числами, десятичная запись которых может содержать до 255 цифр:

а) сложение двух длинных чисел;
б) умножение двух длинных чисел;
в) возведение длинного числа в натуральную степень;
г) вычисление факториала натурального числа;
д) деление длинного числа на длинное число с остатком.

10.22. Реализовать основные операции с вещественными многочленами вида $P(x) = p_n x^n + p_{n-1} x^{n-1} + \cdots + p_1 x + p_0$: сложение, разность, произведение многочленов, производная многочлена, значение многочлена в точке.

10.23. Решить задачу 10.22 для многочленов, коэффициенты, аргументы и значения которых являются комплексными числами.

10.24. Решить задачу 10.22 для многочленов, коэффициенты, аргументы и значения которых являются рациональными числами с выделенной целой частью и дробной частью в виде правильной несократимой обыкновенной дроби.

10.25. Решить задачу 10.22 для многочленов, коэффициенты, аргументы и значения которых являются длинными целыми числами (см. задачу 10.21).

10.26. Реализовать основные операции (сложение, умножение на константу, скалярное произведение) с векторами, элементы которых являются комплексными числами.

10.27. Реализовать основные операции (сложение, умножение на константу, скалярное произведение) с векторами, элементы которых являются рациональными числами с выделенной целой частью и дробной частью в виде правильной несократимой обыкновенной дроби.

10.28. Реализовать основные операции (сложение, умножение на константу, скалярное произведение) с векторами, элементы которых являются длинными целыми числами (см. задачу 10.21).

10.29. Реализовать основные операции (сложение, умножение на константу, умножение на матрицу, транспонирование) с квадратными матрицами, элементы которых являются комплексными числами.

10.30. Реализовать основные операции (сложение, умножение на константу, умножение на матрицу, транспонирование) с квадратными матрицами, элементы которых являются рациональными числами с выделенной целой частью и дробной частью в виде правильной несократимой обыкновенной дроби.

10.31. Реализовать основные операции (сложение, умножение на константу, умножение на матрицу, транспонирование) с квадратными матрицами, элементы которых являются длинными целыми числами (см. задачу 10.21).

11. Файлы

В решениях задач раздела для хранения входных и/или выходных данных предполагается использовать внешние файлы. Файлы могут быть типизироваными,

то есть состоять из компонент определенного типа, или текстовыми. В любом случае, чтение и запись компонент файла производится последовательно. В типизированном файле номер начальной компоненты будем считать равным нулю.

А

11.1. Дан файл f_1, компонентами которого являются вещественные числа. Переписать в файл f_2 те числа из f_1, которые больше среднего арифметического всех чисел f_1.

11.2. Дан файл f_1, компонентами которого являются вещественные числа. Переписать в файл f_2 те числа из f_1, которые находятся после первого из всех минимальных чисел.

11.3. Дан файл f_1, компонентами которого являются вещественные числа. Найти наибольшую из компонент, имеющих четные номера и сумму всех компонент с нечетными номерами.

11.4. Дан файл f_1, компонентами которого являются целые числа. Указать номера тех компонент файла, которые имеют наибольшее или наименьшее значения.

11.5. Даны файлы f_1, f_2, компонентами которых являются целые числа. Покомпонентно поменять местами содержимое этих файлов, используя третий вспомогательный файл.

11.6. Дан файл f_1, компонентами которого являются целые числа. Перенести из файла f_1 в файл f_2 компоненты, которые являются полными квадратами нечетных чисел, исключая их из f_1.

11.7. Дан файл f_1, компонентами которого являются целые числа. Перенести из файла f_1 в файл f_2 компоненты, которые являются числами Фибоначчи, исключая их из f_1 (см. задачу 3.20).

11.8. Дано натуральное число n. В файл f последовательно записать цифры $1, 2, 5, 1, 3, 3, 4, \ldots$, являющиеся цифрами последовательных чисел Фибоначчи, имеющих нечетные номера и не превосходящие n (см. задачу 3.20).

11.9. Дан файл f_1, компонентами которого являются целые числа. Перенести из файла f_1 в файл f_2 компоненты, которые являются степенями $2, 3$ или 5, исключая их из f_1.

11.10. В файл, компонентами которого являются вещественные числа записать все члены конечной последовательности из задачи а) 4.9; б) 4.10.

11.11. Даны файлы f_1, f_2, компонентами которых являются целые числа. Переписать компоненты f_1 и f_2 в файл f_3, чередуя их друг с другом. Если один из файлов закончится раньше другого, то остаток целиком записать в f_3.

11.12. В целочисленном файле f первую компоненту поменять местами с предпоследней.

11.13. Дан файл f_1, компонентами которого являются целые числа. Скопировать в файл f_2 те числа из f_1, которые находятся в файле между первым и последним из экстремальных чисел f_1 (экстремальным называется минимальное или максимальное число).

11.14. Дан файл f_1, компонентами которого являются целые числа. Скопировать компоненты из f_1 в f_2, меняя местами каждые две соседние компоненты с номерами $2k$ и $2k+1$, $(k = 0, 1, 2, \ldots)$.

11.15. Даны файлы f_1, f_2, компонентами которых являются целые числа. Скопировать в файл f_3 наиболее длинную совпадающую начальную последовательность компонент файлов f_1 и f_2.

11.16. Даны файлы f_1, f_2, компонентами которых являются целые числа. Логической переменной b присвоить значение «истина» или «ложь» в зависимости от того, совпадает содержимое файлов f_1 и f_2 или нет. Содержимое совпадает, если длины файлов равны и совпадают компоненты файлов с одинаковыми номерами. Указание: не всегда есть необходимость просматривать содержимое файлов до конца.

11.17. Дан файл f_1, компонентами которого являются однобайтовые целые числа. Скопировать содержимое этого файла в файл f_2, исключая повторные вхождения одних и тех же чисел. Указание: использовать тип «множество».

11.18. В текстовом файле f_1 последовательно размещаются следующие величины: целое число n, элементы вещественной квадратной матрицы A порядка n, элементы вещественного вектора x размера n. В текстовый файл f_2 вывести те же величины, кроме вектора, вместо которого записать элементы вектора y, равного Ax. Затем применить тот же алгоритм $(n-1)$ раз, меняя ролями f_1 и f_2.

11.19. В текстовом файле f_1 две строки, причем во второй строке присутствует символ точки. Записать в файл f_2 первую строку файла f_1, заменив первое вхождение в нее подстроки, совпадающей с частью второй строки до точки на подстроку, следующую после точки. Например, если содержимое файла f_1 такое:

```
Мы не рабы, рабы не мы.
абы.ыба
```

то файл f_2 должен быть таким:

```
Мы не рыба, рабы не мы.
```

11.20. Дан текстовый файл f_1. Скопировать его содержимое в файл f_2, которое следует после слов «`Горшочек, вари`» до конца файла. Если такого текста нет в исходном файле, то в файл f_2 записать текст «`Горшочек, не вари`». Примечание: указанный текст может располагаться в f_1 не только в одной строке, но и в соседних последовательных строках.

11.21. Дан текстовый файл f_1. Скопировать его содержимое в файл f_2 так, чтобы каждая строка, кроме, возможно, последней, содержала бы 60 символов; никакие новые символы включать в текст не следует.

В

11.22. Даны файлы f_1, f_2, компонентами которых являются целые числа. Дописать в конец файла f_1 без повторов те числа, которые имеются в f_2, но отсутствуют в f_1 и, наоборот, в f_2 дописать числа, которые имелись в f_1, но отсутствовали в f_2.

11.23. Даны файлы f_1, f_2, компонентами которых являются целые числа. Записать в файл f_3 числа, которые присутствуют в обоих данных файлах.

11.24. Дан файл целых чисел f_1. Скопировать все компоненты f_1 в файл f_2, поменяв их порядок на обратный. Решить эту задачу:

а) с использованием массива;
б) без использования дополнительных структур.

11.25. Дан файл целых чисел f_1. Все отрицательные числа из f_1 скопировать в файл f_2, сохранив их взаимное расположение, а неотрицательные – в файл f_3, поменяв их порядок на обратный. Решить эту задачу:

а) с использованием массива;
б) без использования дополнительных структур.

11.26. Даны файлы f_1, f_2, компонентами которых являются целые числа. Записать в файл f_3 числа файла f_1, чередуя их с числами файла f_2, взятыми в обратном порядке. Решить эту задачу:

а) с использованием массива;
б) без использования дополнительных структур.

11.27. Слияние упорядоченных файлов. Считая, что компоненты двух целочисленных файлов f_1 и f_2 упорядочены по неубыванию, слить их в файл f_3, также упорядоченный по неубыванию.

11.28. Решить задачу 11.27 для трех исходных упорядоченных файлов.

11.29. Дано натуральное число n. В текстовый файл вывести *треугольник Паскаля*, состоящий из n строк, в следующем виде:

$$\text{а)}\ \begin{array}{llll} 1 & & & \\ 1 & 1 & & \\ 1 & 2 & 1 & \\ 1 & 3 & 3 & 1 \\ \dots & \dots & & \end{array} \qquad \text{б)}\ \begin{array}{ccccccc} & & & 1 & & & \\ & & 1 & & 1 & & \\ & 1 & & 2 & & 1 & \\ 1 & & 3 & & 3 & & 1 \\ & & \dots & \dots & \dots & & \end{array}$$

11.30. В текстовом файле представить алгоритмы сложения, умножения столбиком и деления с остатком двух данных натуральных чисел.

11.31. В текстовом файле f заданы вещественные коэффициенты и целые неотрицательные показатели степеней многочлена $P(x)$, причем пары (коэффициент, показатель) упорядочены по убыванию показателей. Дано вещественное x. Найти значение многочлена $P(x)$ по схеме Горнера (см. задачу 3.2).

11.32. В текстовом файле f_1 находится информация о нескольких студентах: указывается фамилия, инициалы студента, номер группы, дата рождения, оценка. В текстовый файл f_2 вывести список студентов той группы, в которой средняя оценка максимальна.

C

11.33. Дан текстовый файл f_1. Построчно скопировать его содержимое в файл f_2 в порядке невозрастания длин строк.

11.34. Дан текстовый файл f_1. Переписать построчно содержимое этого файла в файл f_2, выравнивая строки:

а) по правому краю;

б) по центру;

в) по ширине (за счет равномерного добавления пробелов между словами добиться того, чтобы строки имели одинаковую длину, равную количеству символов самой длинной строки исходного файла).

11.35. Сортировка файла сбалансированным слиянием. Алгоритм сортировки исходного файла f_1 сбалансированным слиянием предусматривает применение еще трех вспомогательных файлов f_2, f_3, f_4. Для пояснения алгоритма введем понятие *отрезка* – части последовательности наибольшей длины, упорядоченной по неубыванию. Например, в последовательности $\underline{2,\ 5}$, $\underline{3,\ 6,\ 10}$, $\underline{7}$, $\underline{4,\ 7,\ 9}$, $\underline{1}$ пять отрезков и они подчеркнуты. Алгоритм состоит из двух этапов. На первом этапе из исходного файла f_1 отрезки поочередно переносятся в файлы f_3 и f_4. Например, если в исходном файле f_1 была последовательность $\underline{2,\ 5}$, $\underline{3,\ 6,\ 10}$, $\underline{7}$, $\underline{4,\ 7,\ 9}$, $\underline{1}$, то после первого этапа состояние файлов будет таким:

$f_1 :$

$f_2 :$

$f_3 : \underline{2\ 5\ 7}\ \underline{1}$

$f_4 : \underline{3\ 6\ 10}\ \underline{4\ 7\ 9}$

На втором этапе проводят слияние начальных отрезков каждого из двух файлов f_3, f_4 (см. задачу 11.27) поочередно то в файл f_1,то в f_2; как только файлы f_3 и f_4 опустеют, слияние продолжается из файлов f_1, f_2 обратно в файлы f_3, f_4 и т.д. В результате на каком-то очередном шаге окажется, что все числа находятся в одном файле. Его содержимое следует перенести в файл f_1. Продолжим пример:

$f_1 : \underline{2\ 3\ 5\ 6\ 7\ 10}$

$f_2 : \underline{1\ 4\ 7\ 9}$

$f_3 :$

f_4 :
и далее
f_1 :
f_2 :
f_3 : $\underline{1\ 2\ 3\ 4\ 5\ 6\ 7\ 7\ 9\ 10}$
f_4 :
Содержимое f_3 перенесем в исходный файл f_1 после чего сортировка закончена:
f_1 : $\underline{1\ 2\ 3\ 4\ 5\ 6\ 7\ 7\ 9\ 10}$.
Реализовать указанный алгоритм.

12. Рекурсия

При решении задач данного раздела необходимо использовать рекурсивные алгоритмы. Рекурсивным называется алгоритм, в процессе исполнения которого происходит обращение к этому же алгоритму, как правило, с другими значениями входных параметров. Следует отметить, что многие задачи этого раздела могут быть эффективно решены и без привлечения рекурсии, однако настоятельно рекомендуем найти именно рекурсивную версию решения вместе с версией, в основе которой лежит итерация. Исследование, сравнение этих двух версий решения также представляет собой интересную задачу.

А

12.1. Функция *факториал* определяется как итеративно:

$$n! = 1 \cdot 2 \cdot \dots \cdot n,$$

так и рекурсивно:

$$n! = \begin{cases} 1, & \text{если } n = 0; \\ n \cdot (n-1)!, & \text{если } n \geqslant 1. \end{cases}$$

Реализовать рекурсивную функцию, вычисляющую факториал.

12.2. Функция *степень с основанием* $a \neq 0$ *и целым неотрицательным показателем* определяется как итеративно:

$$a^n = \underbrace{a \cdot a \cdot \dots \cdot a}_{n},$$

так и рекурсивно:

$$a^n = \begin{cases} 1, & \text{если } n = 0; \\ a \cdot a^{n-1}, & \text{если } n > 0. \end{cases}$$

Реализовать рекурсивную функцию, вычисляющую степень с целым неотрицательным показателем.

12.3. Функция *степень с основанием* $a \neq 0$ *и целым показателем* определяется следующим образом:

$$a^n = \begin{cases} 1, & \text{если } n = 0; \\ \dfrac{1}{a^{|n|}}, & \text{если } n < 0; \\ a \cdot a^{n-1}, & \text{если } n > 0. \end{cases}$$

Реализовать рекурсивную функцию, вычисляющую степень с целым показателем.

12.4. Последовательность *чисел Фибоначчи* $a_1, a_2, a_3, \dots$ определяется так:

$$a_n = \begin{cases} 1, & \text{если } n = 1 \text{ или } n = 2; \\ a_{n-1} + a_{n-2}, & \text{если } n > 2. \end{cases}$$

Реализовать рекурсивную функцию, вычисляющую n-е число Фибоначчи.

12.5. *Биномиальный коэффициент* для заданных целых n и m $(0 \leqslant m \leqslant n)$ определяется как итеративно:

$$C_n^m = \frac{n!}{m!(n-m)!},$$

так и рекурсивно:

$$C_n^m = \begin{cases} 1, & \text{если } m = 0 \text{ или } m = n; \\ C_{n-1}^m + C_{n-1}^{m-1}, & \text{если } 0 < m < n. \end{cases}$$

Реализовать рекурсивную функцию, вычисляющую биномиальный коэффициент.

12.6. Реализовать рекурсивную функцию, вычисляющую наибольший общий делитель двух целых неотрицательных чисел с помощью алгоритма Евклида (см. определение в задаче 4.14).

12.7. Написать рекурсивную подпрограмму, вводящую с клавиатуры последовательность вещественных положительных чисел, заканчивающуюся нулем, которая возвращает их сумму. Ограничение: массивы, циклы не использовать.

12.8. Написать рекурсивную подпрограмму, вводящую из файла последовательность вещественных чисел, которая выдает на экран вначале отрицательные, а затем – неотрицательные члены последовательности (в любом порядке). Ограничение: массивы, циклы не использовать.

12.9. Дан линейный массив. Реализовать рекурсивную функцию, печатающую элементы массива по порядку. Ограничение: циклы не использовать.

12.10. Дан линейный массив. Реализовать рекурсивную функцию, печатающую элементы массива в обратном порядке. Ограничение: циклы не использовать.

12.11. Дана символьная строка. Реализовать рекурсивную функцию, печатающую символы строки в обратном порядке. Ограничение: циклы не использовать.

12.12. Дана символьная строка. Реализовать логическую рекурсивную функцию, проверяющую, является ли строка палиндромом. Ограничение: циклы не использовать.

12.13. Дан линейный вещественный массив. Реализовать рекурсивную функцию, находящую индекс первого из всех минимальных элементов массива. Ограничение: циклы не использовать.

12.14. Даны две символьные строки s_1, s_2. Реализовать рекурсивную функцию, вычисляющую количество вхождений строки s_2 в строку s_1. Ограничение: циклы не использовать.

12.15. Дано целое неотрицательное число n. Реализовать рекурсивную функцию, вычисляющую количество и сумму цифр в десятичной записи n. Ограничение: циклы не использовать.

12.16. Функция $f(n)$ определена для натуральных n следующим образом:

$$f(n) = \begin{cases} 1, & \text{если } n = 1; \\ \sum\limits_{i=2}^{n} f(n\ div\ i), & \text{если } n > 1. \end{cases}$$

Построить таблицу значений $f(n)$ для $n = 1, 2, \ldots, 30$.

В

12.17. *Функция Аккермана* для целых неотрицательных n, m определяется следующим образом:

$$A(n, m) = \begin{cases} m + 1, & \text{если } n = 0; \\ A(n-1, 1), & \text{если } n \neq 0 \text{ и } m = 0; \\ A(n-1, A(n, m-1)), & \text{в остальных случаях.} \end{cases}$$

Реализовать функцию Аккермана. Исследовать зависимость времени вычисления функции от значений аргументов.

12.18. Реализовать решение уравнения $f(x) = 0$ на отрезке $[a; b]$ с точностью $\varepsilon > 0$ методом деления отрезка пополам с помощью рекурсивной функции (считать, что $f(x)$ – непрерывная монотонная функция на $[a; b]$, $a < b$, $f(a) \cdot f(b) < 0$). На каждом шаге отрезок делится пополам и из двух половин выбирается та, на концах которой функция $f(x)$ принимает значения разных знаков; этот процесс продолжается до тех пор, пока длина очередного отрезка не станет меньше ε или не будет найдено решение.

12.19. Вычислить значения математических функций из задачи 4.11 по их разложениям в ряд. Ограничение: циклы не использовать.

12.20. Решить задачу 3.45 с помощью рекурсии.

12.21. Задача о Ханойской башне. Древняя легенда гласит, что в буддистском храме имеется три стержня (назовем их для простоты A, B и C); на стержне A нанизаны n дисков разного диаметра в виде пирамиды. Монахи могут переносить диски с одного стержня на другой, соблюдая при этом правила: в любой момент времени снят может быть только один диск и диск большего диаметра нельзя ставить на диск меньшего диаметра. Легенда утверждает, что как только монахи перенесут все диски со стержня A на стержень C, наступит конец света (по легенде $n = 64$). Написать программу, печатающую последовательность переноса n дисков, каждое действие должно иметь вид «перенести диск со стержня X на стержень Y», где X и Y – это A, B или C.

C

12.22. Алгоритм быстрой сортировки. Дан линейный вещественный массив $a_1, a_2, \ldots, a_n$. Алгоритм быстрой сортировки заключается в следующем: из массива выбирают один из элементов a_k (обычно – средний элемент), затем, двигаясь от начала массива слева направо, находят первый элемент, больший a_k; двигаясь от конца массива справа налево, находят первый элемент, не больший a_k; найденные элементы меняют местами. Движение слева и движения справа повторяют, чтобы найти следующую пару элементов для обмена; в последнюю очередь занимает свое окончательное место элемент a_k. В результате массив будет состоять из двух групп элементов: элементы слева от a_k не большие его, а справа – элементы, превосходящие a_k. Затем рекурсивно применяют алгоритм к левой и правой частям массива. Реализовать рекурсивный алгоритм быстрой сортировки массива.

12.23. Реализовать рекурсивную версию алгоритма сортировки выбором (см. задачу 5.27).

12.24. Реализовать рекурсивную версию алгоритма сортировки вставками (см. задачу 5.28).

12.25. Реализовать рекурсивную версию алгоритма сортировки обменом (см. задачу 5.29).

12.26. Имеется N населенных пунктов. Некоторые пары пунктов соединены дорогами. Требуется определить, можно ли из пункта с номером n_1 попасть в пункт с номером n_2 и, если можно, то указать какой-нибудь путь. Система дорог задается в виде квадратной симметричной относительно главной диагонали матрицы, состоящей из нулей и единиц: если элемент a_{ij} равен 1, значит пункты с номерами i и j напрямую соединены дорогой, а если равен 0 – то не соединены (матрица смежности). Путь нужно указать в виде последовательности номеров пунктов $i_1, i_2, \ldots, i_k$, где $i_1 = n_1, i_k = n_2$.

12.27. Имеется N населенных пунктов. Некоторые пары пунктов соединены дорогами определенной длины. Требуется определить, можно ли из пункта с номером n_1 попасть в пункт с номером n_2 и, если можно, то каков наименьший путь. Система дорог задается в виде квадратной симметричной относительно главной диагонали матрицы, состоящей из вещественных неотрицательных чисел: если элемент $a_{ij} = L$ больше 0, значит пункты с номерами i и j напрямую соединены дорогой длины L, а если равен 0 – то не соединены. Путь нужно указать в виде последовательности номеров пунктов $i_1, i_2, \ldots, i_k$, где $i_1 = n_1, i_k = n_2$.

12.28. Найти все такие расстановки восьми ферзей на шахматной доске, чтобы ферзи не били друг друга.

12.29. Лошадью ходи! Ходом коня обойти всю шахматную доску, начиная с клетки $a1$, не проходя по каждой клетке более одного раза.

12.30. Лошадью ходи!-2. Даны координаты двух клеток шахматной доски. Найти наиболее короткий из путей от первой до второй клетки, двигаясь ходом коня. В каждой клетке можно побывать не более одного раза.

12.31. Без параметров... Напишите рекурсивную версию функции, вычисляющей факториал целого положительного числа (см. задачу 12.1)

а) без использования параметров;

б) без использования параметров и локальных данных.

Каковы трудности и недостатки такого подхода?

12.32. Дана вещественная квадратная матрица a порядка n. Найти определитель матрицы A, используя формулу разложения по k-й строке:

$$\det A = \sum_{i=1}^{n} (-1)^{k+i} a_{ki} \cdot \det A_{ki} \ ,$$

где A_{ki} – матрица (минор), полученная из матрицы A выбрасыванием k-й строки и i-го столбца. Указание: реализовать рекурсивную функцию от двух параметров-множеств s_1 и s_2, вычисляющую по данной формуле определитель матрицы, полученной из исходной выбрасыванием строк, номера которых принадлежат s_1, и столбцов, номера которых принадлежат s_2.

12.33. Города. Дан список русских названий городов. Требуется построить максимально длинную цепочку названий городов таким образом, чтобы первая буква названия каждого следующего города совпадала с последней буквой названия предыдущего города. Если название города заканчивается на букву "ь", то она игнорируется. Город в цепочку может входить только один раз. Вместо названий городов можно взять любой список слов.

12.34. Дорожная разметка. Система городских улиц состоит из перекрестков и соединящих их дорог. Количество перекрестков — n, они пронумерованы от 1 до n, каждая дорога обозначается номерами двух перекрестков i и j, которые она соединяет, a_{ij} — длина дороги. Разметочная машина наносит на дорожное полотно разметку, двигаясь по дорогам от перекрестка к перекрестку, при этом она может идти в двух режимах: с покраской и без покраски. Требуется определить такой маршрут движения машины, чтобы разметка была нанесена на все дороги и при этом пройденный ею путь был бы минимальным. Путь машины начинается от перекрестка с номером k. Входные данных из файла задаются в следующей последовательности:

```
<Количество перекрестков n>
<Начальный перекресток k>
<Количество дорог m>
<№ перекр.-начало><№ перекр.-конец><Длина дороги 1>
<№ перекр.-начало><№ перекр.-конец><Длина дороги 2>
. . .
<№ перекр.-начало><№ перекр.-конец><Длина дороги m>.
```

Выходные данные определяют маршрут движения машины и режим работы на каждой дороге; выводятся построчно по три числа в каждой строке: номер перекрестка-начало дороги, номер перекрестка-конец дороги, режим движения (0 — разметка не наносится, 1 — разметка наносится).

12.35. Почти по Маркову. Даны n правил преобразования строк в виде пар $(\alpha_1, \beta_1), (\alpha_2, \beta_2), \ldots, (\alpha_n, \beta_n)$, где α_i и β_i — некоторые строки, α_i— непустая строка. Правило (α_i, β_i) называется применимым к строке x, если α_i является подстрокой строки x. Применением применимого правила (α_i, β_i) к строке x называется замена первого вхождения подстроки α_i в строку x на подстроку β_i. Например, при применении к строке "abcabcd" правила ("bc","klm") получается строка "aklmabcd".

Даны две строки s и f, правила преобразования. Требуется указать наиболее короткую последовательность применения правил к строке s для того, чтобы получить строку f или сообщить, что этого сделать невозможно. Правила указываются своими номерами.

Пример.

(Здесь кавычки не являются частью строк.)
Строка s=*МУХА*
Строка f=*СЛОН*
Правила:*("МУХ","СЛО"),("А","НН"),("Н",""),*
("МУХ","СЛ"), ("ХА","ХОН")
Ответ: номера примененных правил – 5, 4.

13. Динамические структуры данных. Линейные списки

В решениях задач этого раздела необходимо использовать динамические структуры данных – линейные одно- или двусвязные списки (последовательности значений). Следует заметить, что реорганизация списка (включение, исключение, перестановка элементов) должна осуществляться измением ссылочных, а не информационных полей элементов списка. Кроме того, считается, что перед выполнением любой операции со списком его длина (количество элементов) явно не за-

дана.

А

13.1. Дан список из n целых неотрицательных чисел $a_1, a_2, \ldots, a_n$. Найти:

а) $a_1 + a_2 + \cdots + a_n$;

б) $(a_1 + a_2) \cdot (a_2 + a_3) \cdot \cdots \cdot (a_{n-1} + a_n)$;

в) $(a_1a_2)\ mod\ a_3 + (a_2a_3)\ mod\ a_4 + \cdots + (a_{n-2}a_{n-1})\ mod\ a_n$.

13.2. Дан список из n целых неотрицательных чисел $a_1, a_2, \ldots, a_n$. Найти:

а) $a_1 - a_2 + \cdots \pm a_n$;

б) $a_1a_2 + a_2a_3 + \cdots + a_{n-1}a_n$;

в) $(a_1+a_2)\ div\ a_n + (a_2+a_3)\ div\ a_{n-1} + \cdots + (a_{n-1}+a_n)\ div\ a_1$.

13.3. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Вывести на экран те элементы списка, которые больше своих соседей.

13.4. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Вывести на экран элементы списка в указанной последовательности:

а) $a_n, a_{n-1}, \ldots, a_1$;

б) $a_1, a_n, a_2, a_{n-1}, \ldots, a_n, a_1$;

в) $a_1, a_2, \ldots, a_{\frac{n}{2}}, a_n, a_{n-1}, \ldots, a_{\frac{n}{2}+1}$, ($n$–четное).

13.5. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Вывести на экран элементы списка в указанной последовательности:

а) $a_1, a_3, a_2, a_4, a_5, \ldots, a_n$;

б) $a_n, a_1, a_{n-1}, a_2, \ldots, a_1, a_n$;

в) $a_2, a_1, a_4, a_3, \ldots, a_n, a_{n-1}$, ($n$–четное).

13.6. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Исключить из списка элементы с минимальным значением.

13.7. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Включить после каждого элемента с максимальным значением его копию (продублировать элемент).

13.8. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Поменять местами первый из неотрицательных и последний из максимальных элементов списка.

13.9. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Поменять местами первый из минимальных и последний из положительных элементов списка.

13.10. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Поменять местами первый элемент и элемент, значение которого наиболее близко к среднему арифметическому всех элементов списка.

13.11. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Удалить все элементы, равные полусумме своих соседей в исходном списке.

13.12. Дан список из $n \geqslant 2$ целых чисел $a_1, a_2, \ldots, a_n$, среди которых по крайней мере два различны. Перед каждым элементом списка, имеющим наименьшее значение, вставить элемент с наибольшим значением.

13.13. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Переместить в начало все отрицательные элементы списка, сохранив их взаимное расположение.

13.14. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Последовательно переместить все отрицательные элементы списка в его начало, а все положительные – в его конец.

13.15. Дан двусвязный список из n целых чисел $a_1, a_2, \ldots, a_n$. Перевернуть список, то есть изменить ссылки так, чтобы его элементы оказались расположенными в обратном порядке.

13.16. Решить задачу 13.15 для односвязного списка.

13.17. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Из группы подряд идущих элементов с одинаковыми значениями оставить только первый.

13.18. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Исключить из списка все повторные вхождения элементов.

13.19. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Определить, является ли он симметричным.

13.20. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Удалить из списка все элементы, входящие в него в точности два раза.

13.21. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Удалить все элементы списка, находящиеся между первым из всех минимальных и последним их всех максимальных элементов.

13.22. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. «Перевернуть» ту часть элементов списка, которая находится между первым из всех отрицательных и последним из всех неотрицательных элементов списка (считать, что такие элементы в списке обязательно существуют).

13.23. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Переместить в начало списка все элементы, которые являются локальными максимумами, то есть больше своих соседей.

13.24. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Найти самую длинную неубывающую подпоследовательность элементов списка.

13.25. Дан список из n целых чисел $a_1, a_2, \ldots, a_n$. Удалить все пары соседних элементов исходного списка, значения которых отличаются не более, чем на $C > 0$.

13.26. Даны два списка L_1 и L_2. Включить после каждого элемента списка L_1 с макимальным значением последовательность элементов списка L_2.

13.27. Слияние упорядоченных списков в третий. Даны два списка L_1 и L_2. Считая, что элементы списков упорядочены по неубыванию, создать третий список L_3 из элементов L_1 и L_2 так, чтобы он тоже был упорядочен по неубыванию.

13.28. Слияние упорядоченных списков вставкой. Даны два списка L_1 и L_2. Считая, что элементы списков упорядочены по неубыванию, вставить элементы списка L_2 в список L_1 так, чтобы он остался упорядочен по неубыванию.

13.29. Считалка, задача Иосифа. По кругу становятся n человек, пронумерованных от 1 до n. Начиная отсчет от первого человека, удаляют из круга m-го человека, затем многократно повторяют этот процесс до тех пор, пока в кругу не останется последний человек. Даны натуральные n, m. Найти номер человека, который останется последним в кругу после применения m-считалки для n человек.

В

13.30. Даны натуральные n, k ($k \leqslant n$). Найти такое натуральное m, чтобы после применения m-считалки (см. задачу 13.29) к n людям последним в кругу остался бы человек с номером k или сообщить, что такого m не существует.

13.31. Даны три целочисленных списка L_1, L_2 и L_3. Заменить каждое вхождение списка L_2 в список L_1 на список L_3.

13.32. Даны два целочисленных списка L_1 и L_2. Построить новый список L_3, включив в него элементы, которые:

а) входят одновременно в L_1 и в L_2;
б) входят хотя бы в один из списков L_1, L_2;
в) входят в L_1, но не входят в L_2;
г) входят ровно в один из списков L_1, L_2.

13.33. Дан список вещественных элементов. Провести сортировку элементов списка методом выбора (см. задачу 5.27).

13.34. Дан список вещественных элементов. Провести сортировку элементов списка методом обмена – пузырьковую сортировку (см. задачу 5.29).

13.35. Дан список вещественных элементов. Провести сортировку элементов списка методом вставки (см. задачу 5.28).

13.36. Дан список вещественных элементов. Провести быструю сортировку элементов списка (см. задачу 12.22).

13.37. Дан список вещественных элементов. Провести сортировку элементов списка сбалансированным слиянием (см. задачу 11.35).

13.38. Многочлен n-й степени можно представить в виде списка, где каждый элемент состоит из двух чисел – ненулевого коэффициента и показателя степени. Описать соответствующий тип данных и реализовать на нем следующие операции:

а) ввод многочлена с клавиатуры или из файла;
б) приведение многочлена к стандарному виду (приведены подобные слагаемые и одночлены упорядочены по убыванию показателей степени);
в) вывод многочлена на экран в обычном виде (например, "p(x)=4x^3-2.51x+1.2");
г) проверка равенства двух многочленов;
д) сумма, разность, произведение двух многочленов;
е) нахождение производной многочлена;
ж) деление с остатком многочлена на многочлен;
з) нахождение значения многочлена в точке x по схеме Горнера (см. задачу 3.2).

13.39. Стек (stack, магазин). Реализовать структуру данных – стек. Стек представляет собой линейный список, доступ к элементам которого разрешен лишь с одного конца списка, называемого вершиной стека. На структуре стек реализовать основные операции: `push` – включение элемента в стек (то есть на вершину стека); `pop` – взятие элемента с вершины стека; `empty` – проверка стека на пустоту; `clear` – очистка стека. Программа должна по командам(push, pop, empty, clear) выполнять соответствующие операции над стеком.

13.40. Очередь (queue). Реализовать структуру данных – очередь. Очередь представляет собой линейный список, включение новых элементов в который осуществляется на одном конце (конец очереди), а исключение – на другом конце (начало очереди). На структуре очередь реализовать основные операции: `put` – включение нового элемента в очередь; `get` – взятие элемента из очереди; `empty` – проверка на пустоту, `clear` – очистка очереди. Программа должна по командам (put, get, empty, clear) выполнять соответствующие операции над очередью.

13.41. Дек (deque). Реализовать структуру данных – дек. Дек представляет собой линейный список, в котором включение новых элементов и исключение элементов производится на двух концах – левом и правом. На структуре дек реализовать основные операции: `putleft/putright` – включение нового элемента в левый/правый конец дека, `getleft/getright` – взятие элемента из левого/правого конца дека; `empty` – проверка на пустоту, `clear` – очистка дека. Программа должна по командам (putleft, putright, getleft, getright, empty, clear) выполнять соответствующие операции над деком.

C

13.42. Железнодорожный сортировочный узел представляет собой три тупиковых ветки A, B и C, соединенные в одной точке. Из каждой ветки в другую можно перегонять вагоны. Состав находится на ветке A. В составе находятся вагоны двух типов – первого и второго, каждый вагон имеет инвентарный номер. Распределить вагоны по веткам B и C так, чтобы на ветке B находились вагоны первого типа, а на C – второго. Указать последовательность инвентарных номеров вагонов в двух образованных составах, начиная с последнего, присоединенного к каждому составу. Использовать в программе стек (см. задачу 13.39).

13.43. В железнодорожных кассах происходит продажа билетов пассажирам. Приходящий пассажир выбирает ту кассу, где очередь минимальна, и становится в ее конец. Пассажир, купивший билет, покидает очередь. Смоделировать процесс работы касс, используя структуру очереди (см. задачу 13.40). Процесс работы представить как последовательность событий (приход пассажира X и постановка его в очередь кассы Y, уход пассажира X из очереди), каждое событие характеризуется временем (часы, минуты). Считать, что время прихода очередного пассажира – величина случайная, время обслуживания пассажира в кассе колеблется от 1 до 10 минут.

13.44. Дан некоторый текст, в котором особую роль играет символ # – он означает отмену одного предыдущего символа. Несколько подряд идущих символов # отменяют такое же количество предыдущих символов текста. Например, если задан текст `"Н#НЕ##ПРЮ#ИВЕТТ#"`, то после преобразования он превратится в `"ПРИВЕТ"`. Преобразовать входной текст с учетом роли знака #. Указание: использовать стек (см. задачу 13.39).

13.45. Обратная польская (постфиксная) запись. Запись бинарной операции, когда знак операции записывается после обоих операндов, называется *обратной польской записью* (ОПЗ). Например, операция "2 3 +" в ОПЗ означает в привычной (инфиксной) записи "$2+3$", а выражение "1 2 3 $*$ +" означает "$1+2*3$". Во входной строке дано арифметическое выражение в обратной польской записи; в выражение могут входить знаки арифметических операций ($+$, $-$, $*$, $/$) и целые числа, причем два рядом стоящих целых числа отделяются друг от друга по крайней мере одним пробелом. Найти значение выражения. Указание: использовать стек (см. задачу 13.39).

13.46. Решить задачу 13.45 для обратной польской записи логического выражения. В логическое выражение могут входить константы T (истина) и F (ложь), бинарные операции & (И), v (ИЛИ), x (исключающее ИЛИ), унарная операция ˜ (НЕ).

Указание: использовать стек (см. задачу 13.39).

13.47. Дан текст, сбалансированный по круглым скобкам. Указать номера позиций в тексте для каждой пары открывающей и закрывающей скобок. Например, для текста "`pa(sc(a()l))`"пары следующие: 3 и 12, 6 и 11, 8 и 9. Указание: использовать стек (см. задачу 13.39).

13.48. В текстовом файле находятся строки текста, разбитого на отдельные слова. Слова отделяются друг от друга знаками препинания или пробелами. Представить каждую строку как список слов, а весь текст – как список строк. Найти все слова, которые входят в каждую строку; входят в какую-либо строку не менее двух раз; входят в весь текст только один раз.

13.49. (Задача из [3]). Определим понятие *иерархический список* как заключенную в круглые скобки последовательность элементов, разделенных запятыми; каждый элемент – либо атом (идентификатор), либо иерархический список. Написать программу, осуществляющую ввод иерархического списка (списков), проверку принадлежности атома списку, вывод списка на экран, проверку равенства двух списков.

13.50. Решить задачу о Ханойской башне 12.21 с применением стека без использования рекурсивных подпрограмм.

13.51. Решить задачи 12.26, 12.27 о связанных населенных пунктах с применением стека без использования рекурсивных подпрограмм.

13.52. Решить задачу о расстановке восьми ферзей 12.28 с применением стека без использования рекурсивных подпрограмм.

14. Бинарные деревья, графы

При решении задач раздела необходимо использовать динамические структуры данных для моделирования бинарных (двоичных) деревьев. Для успешного выполнения программ введем здесь ряд понятий, встречающихся в формулировках задач.

Бинарное (двоичное) дерево **– конечное множество вершин с их значениями, которое или пусто, или состоит из корневой вершины и двух бинарных поддеревьев – левого и правого. Поддеревья со своими вершинами называются** ***потомками*** **корня. Вершины, не имеющие потомков, называются** ***листьями. Уровнем*** **вершины называется целое неотрицательное число, равное количеству вершин, проходимых от данной вершины до корня; уровень корня равен нулю.**

Бинарное дерево поиска **– бинарное дерево, значение в каждой вершине которого больше значений ее левого поддерева и меньше значений ее правого поддерева (если эти поддеревья имеются).**

Бинарное дерево выражения **представляет структурные характеристики выражений, построенные на основе бинарных операций, например – арифметических. В корне дерева выражения находится операнд (тогда поддеревьев нет) или знак операции (в этом случае поддеревья содержат выражения-операнды).**

Существует три направления обхода бинарного дерева: прямой (вначале «посещается» корень, затем левое поддерево, затем – правое), обратный (левое поддерево, корень, правое поддерево) и концевой (левое поддерево, правое поддерево, корень). Заметим, что обратный обход бинарного дерева поиска дает возрастающую последовательность значений вершин. Прямой, обратный и концевой обходы бинарного дерева выра-

жения дают соответственно префиксную, инфиксную и постфиксную (обратную польскую) запись выражения.

Отметим, что большинство алгоритмов, в которых использованы деревья, рекурсивны по своей природе, а, значит, при их программировании лучше применять рекурсивные подпрограммы. В большинстве задач раздела не указан источник входных данных для формирования дерева: обучающемуся предлагается самостоятельно определить источник и формат входных данных. Определение формата выходных данных также возлагается на читателя.

A

14.1. Дано целочисленное бинарное дерево. Найти:

а) глубину дерева – максимальный уровень вершины дерева;

б) значение самой правой вершины в левом поддереве от корня;

в) среднее арифметическое значений всех вершин дерева.

14.2. Дано целочисленное бинарное дерево. Найти:

а) количество вершин дерева;

б) значение самой левой вершины в правом поддереве;

в) максимальное и минимальное значение вершин дерева.

14.3. Дано целочисленное бинарное дерево. Найти:

а) количество листьев дерева;

б) количество вершин на n-м уровне;

в) сумму значений всех вершин дерева.

14.4. Дано целочисленное бинарное дерево. Найти:

а) количество вершин дерева, значения которых равны значению корня;

б) количество листьев дерева, находящихся на n-м уровне;

в) сумму значений нелистовых вершин дерева.

14.5. Дано целочисленное бинарное дерево. Выполнить следующее:

а) найти значение, которое имеют наибольшее количество вершин дерева;
б) поменять местами корень и одну из вершин (любую) с наибольшим уровнем;
в) каждой вершине, имеющей единственное поддерево – левое, добавить правое поддерево с вершиной, равной значению левого поддерева.

14.6. Дано целочисленное бинарное дерево. Выполнить следующее:

а) найти сумму всех отрицательных значений вершин дерева и минимальное среди всех положительных значений;
б) поменять местами значения корня и одной из тех вершин, значения которых встречаются в дереве наибольшее число раз;
в) если значение листа, являющегося левым поддеревом, больше, чем значение вершины, которой он подчинен, то их значения поменять местами.

14.7. Дано целое неотрицательное n. Построить бинарное дерево глубины n, корень которого имеет значение 0, а каждая вершина – значение, совпадающее с ее уровнем.

14.8. Дано целое неотрицательное n. Построить бинарное дерево глубиной n, корень которого имеет значение n, а каждая вершина i-го уровня – значение, равное $n - i$.

14.9. Назовем пару различных вершин дерева двойниками, если их значения и уровни совпадают. Найти всех двойников в данном целочисленном бинарном дереве.

14.10. Построение бинарного дерева поиска. На вход подается последовательность целых чисел. Каждое входное число вставить в дерево поиска, для этого в первую очередь необходимо найти место вставки, начиная поиск от корня (если вставляемая вершина меньше, переходим к левому поддереву, если больше – к правому; если поддерево отсутствует – это и будет место вставки), затем производится вставка.

14.11. Поиск в дереве. В данном бинарном дереве поиска найти уровень вершины, значение которой равно a.

14.12. Удаление вершины из дерева поиска. Чтобы удалить вершину d из дерева поиска, нужно воспользоваться следующим алгоритмом: если у d нет потомков, то ее просто удалить; если у d один потомок-поддерево, то d удалить, а потомка переместить на то место, где до этого была вершина d; если у d два потомка, то нужно найти в левом потомке самую правую вершину (или в правом – самую левую) и заместить ею удаляемую вершину d. Дано целочисленное бинарное дерево поиска, целое число d. Удалить из дерева вершину, значение которой равно d, или сообщить, что такой вершины не существует.

B

В задачах 14.13-14.24 по имеющейся информации входного файла необходимо построить бинарное дерево поиска, вершинами которого являются блоки информации, хранимые в файле (поезд, студент, абонент и т. д.). Ключом мы называем такое поле вершины дерева поиска, в соответствии со значениями которого происходит построение дерева.

14.13. Во входном файле находится информация о движении поездов по станции в следующем виде: номер поезда, станция отправления, станция назначения, время прибытия (часы, минуты), время стоянки (минуты), время отправления (часы, минуты). Построить бинарное дерево поиска, в качестве ключа использовать номер поезда. Обеспечить:

а) поиск поезда по его номеру;
б) выдачу всего списка поездов по возрастанию номеров;
в) поиск поезда по станции назначения;

г) поиск поезда по времени отправления;
д) удаление информации о поезде по его номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является время отправления.

14.14. Во входном файле находится информация о расписании полетов самолетов в следующем виде: номер рейса, аэропорт назначения, время вылета (часы, минуты), тип самолета. Построить бинарное дерево поиска, в качестве ключа использовать номер рейса. Обеспечить:
а) поиск рейса по номеру ;
б) выдачу всего списка рейсов по возрастанию номеров;
в) поиск рейса по аэропорту назначения;
г) поиск рейса по времени вылета;
д) удаление информации о рейсе по номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является время вылета.

14.15. Во входном файле находится информация о абонентах телефонной станции в следующем виде: номер телефона, фамилия абонента, адрес абонента (улица, дом, квартира). Построить бинарное дерево поиска, в качестве ключа использовать номер телефона. Обеспечить:
а) поиск абонента по номеру телефона;
б) выдачу всего списка абонентов по возрастанию номеров телефонов;
в) поиск абонента по фамилии;
г) удаление информации о абоненте по номеру телефона;
д) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является адрес абонента.

14.16. Во входном файле находится информация о студентах факультета в следующем виде: фамилия, имя, отчество, номер студенческого билета, номер группы. Построить бинарное дерево поиска, в качестве ключа использовать номер студенческого билета. Обеспечить:

а) поиск студента по номеру студенческого билета;
б) выдачу всего списка студнтов по возрастанию номеров билетов;
в) поиск студента по фамилии, имени, отчеству;
г) поиск студентов по номеру группы;
д) удаление информации о студенте по номеру билета;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: фамилия, имя, отчество.

14.17. Во входном файле находится информация о автомобилях автопарка в следующем виде: государственный номер автомобиля (в стандартном виде), марка автомобиля, фамилия водителя, год и месяц выпуска автомобиля. Построить бинарное дерево поиска, в качестве ключа использовать госномер автомобиля. Обеспечить:
а) поиск автомобиля по госномеру;
б) выдачу всего списка автомобилей по возрастанию госномеров;
в) поиск автомобиля по фамилии водителя;
г) поиск автомобилей по году выпуска;
д) удаление информации об автомобиле по его госномеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: фамилия водителя, год и месяц выпуска автомобиля.

14.18. Во входном файле находится информация о троллейбусах транспортного депо в следующем виде: номер машины, номер маршрута, год выпуска, год и месяц последнего ремонта, фамилия водителя. Построить бинарное дерево поиска, в качестве ключа использовать номер машины. Обеспечить:
а) поиск троллейбуса по номеру машины;
б) выдачу всего списка троллейбусов по возрастанию номеров;
в) поиск троллейбусов по номеру маршрута;
г) поиск троллейбусов по году выпуска;

д) удаление информации о троллейбусе по номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: номер маршрута, фамилия водителя.

14.19. Во входном файле находится информация о книгах в библиотеке в следующем виде: инвентарный номер книги, название, автор, издательство, год, номер стеллажа. Построить бинарное дерево поиска, в качестве ключа использовать инвентарный номер. Обеспечить:

а) поиск книги по инвентарному номеру;
б) выдачу всего списка книг по возрастанию инвентарных номеров;
в) поиск книг по автору;
г) поиск книги по названию;
д) удаление информации о книге по инвентарному номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является название.

14.20. Во входном файле находится информация об ответственных квартиросъемщиках из регистрационной книги домоуправления в следующем виде: номер дома, номер квартиры, фамилия квартиросъемщика, общая площадь квартиры. Построить бинарное дерево поиска, в качестве ключа использовать совокупность: номер дома, номер квартиры. Обеспечить:

а) поиск квартиросъемщика по номеру дома и квартиры;
б) выдачу всего списка квартиросъемщиков;
в) поиск квартиросъемщиков по общей площади квартиры;
г) поиск всех квартиросъемщиков из данного дома;
д) удаление информации о квартиросъемщике по номеру дома и квартиры;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является фамилия квартиросъемщика.

14.21. Во входном файле находится информация по учету материальных ценностей организации в следующем виде: инвентарный номер, название, стоимость, фамилия материально-ответственного лица (МОЛ). Построить бинарное дерево поиска, в качестве ключа использовать инвентарный номер. Обеспечить:

а) поиск материальных ценности по инвентарному номеру;
б) выдачу всего списка материальных ценностей по возрастанию инвентарных номеров;
в) поиск материальных ценностей по фамилии МОЛ;
г) поиск материальных ценностей, стоимость которых превосходит m;
д) удаление информации о материальной ценности по ее инвентарному номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является название.

14.22. Во входном файле находится информация о сотрудниках предприятия в следующем виде: табельный номер, фамилия, имя, отчество, дата рождения (число, месяц, год), адрес. Построить бинарное дерево поиска, в качестве ключа использовать табельный номер. Обеспечить:

а) поиск сотрудника по табельному номеру;
б) выдачу всего списка сотрудников по возрастанию табельных номеров;
в) поиск сотрудника по фамилии, имени, отчеству;
г) поиск сотрудника по адресу;
д) удаление информации о сотруднике по его табельному номеру;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: фамилия, имя, отчество.

14.23. Во входном файле находится информация о песнях в фонотеке в следующем виде: название, композитор, автор слов, исполнитель, год выпуска. Построить бинарное дерево поиска, в качестве ключа использовать название. Обеспечить:

а) поиск песни по названию;
б) выдачу всего списка песен в алфавитном порядке названий;
в) поиск песен конкретного исполнителя;
г) поиск песен с годом выпуска, не превосходящим m;
д) удаление информации о песне по ее названию;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: исполнитель, года выпуска.

14.24. Во входном файле находится информация о кинофильмах в следующем виде: название, киностудия, режиссер, год выпуска. Построить бинарное дерево поиска, в качестве ключа использовать название. Обеспечить:
а) поиск кинофильма по названию;
б) выдачу всего списка кинофильмов в алфавитном порядке названий;
в) поиск кинофильмов конкретного режиссера;
г) поиск кинофильмов с годом выпуска, превосходящим m;
д) удаление информации о кинофильме по его названию;
е) построение по имеющемуся дереву другого бинарного дерева поиска, ключом которого является совокупность: режиссер, год выпуска.

14.25. Во входном текстовом файле задано синтаксически правильное выражение; синтаксис выражения следующий:
```
<выражение> ::= <цифра>|(<цифра><знак><выражение>)
<знак> ::= +|-|*
<цифра> ::= 0|1|2|3|4|5|6|7|8|9
```
Построить по заданному выражению бинарное дерево выражения, вывести его на экран в виде дерева, найти значение выражения.

14.26. Во входном текстовом файле задано синтаксически правильное логическое выражение; синтаксис выражения следующий:

```
<лвыраж> ::= <лконстанта>|(<лконстанта><операция><лвыраж>)
<операция> ::= &|v|x
<лконстанта> ::= T|F
```

Построить по заданному логическому выражению бинарное дерево выражения, вывести его на экран в виде дерева, найти значение выражения.

C

14.27. Во входном текстовом файле задан текст. Проверить, является ли текст правильной записью выражения согласно синтаксису из задачи 14.25 и, если является, построить по нему дерево выражения, вывести дерево на экран, найти значение выражения.

14.28. Во входном текстовом файле задан текст. Проверить, является ли текст правильной записью логического выражения согласно синтаксису из задачи 14.26 и, если является, построить по нему дерево выражения, вывести дерево на экран, найти значение выражения.

14.29. Во входном текстовом файле задан текст. Проверить, является ли текст записью правильного выражения согласно синтаксису:

```
<выражение> ::= <терм>|<терм><знакпм><выражение>
<терм> ::= <целоечисло>|<целоечисло>*<терм>
<знакпм> ::= +|-
<целоечисло> ::= <цифра>|<цифра><целоечисло>
<цифра> ::= 0|1|2|3|4|5|6|7|8|9
```

и, если является, построить дерево выражения, вывести его на экран, вычислить значение выражения.

14.30. Преобразовать выражение, заданное в текстовом файле, согласно синтаксису из задачи 14.29 в инфиксной форме в обратную польскую запись (см. задачу 13.45).

В задачах 14.31 – 14.42 используются графы. Для

представления графа можно воспользоваться матрицей смежности, как это было сделано в задаче 12.26, но также можно построить динамическую структуру связанных элементов – вершин графа. В задачах этого раздела предлагается воспользоваться вторым методом.

Введем здесь основные понятия теории графов. *Граф* представляет собой множество вершин $V = \{v_1, v_2, \ldots, v_n\}$ и множество дуг, соединяющих некоторые пары вершин $E = \{\langle v_i, v_j\rangle | 1 \leqslant i, j \leqslant n\}$; в дуге $\langle v_i, v_j\rangle$ вершина v_i называется началом дуги, v_j – концом дуги. Если вместе с каждой дугой $\langle v_i, v_j\rangle$ в E входит и дуга $\langle v_j, v_i\rangle$, то граф называется *неориентированным*, в противном случае он называется *ориентированным*, или *орграфом*. *Цепью* в графе называется последовательность отличных друг от друга дуг, конец каждой дуги при этом совпадает с началом следующей дуги; начало первой дуги называется началом цепи, конец последней дуги – концом цепи. *Простой цепью* называется цепь, вершины дуг которой не повторяются (то есть простая цепь не проходит через одну вершину дважды). *Замкнутой цепью* называется цепь, начало и конец которой совпадают. *Циклом* называется замкнутая простая цепь, содержащая по крайней мере одну дугу.

Формат входных данных определить самостоятельно, но проще всего графы задавать в виде последовательностей пар натуральных чисел $(a; b)$, где a и b – номера соединенных дугой вершин; вершины идентифицировать натуральными числами. Формат выходных данных также предлагается определить самостоятельно, в том числе допускается графическое представление графа на экране.

14.31. Дан ориентированный граф. Сделать из него неориентированный граф, то есть для каждой дуги $\langle v_i, v_j\rangle$ в множество дуг добавить дугу $\langle v_j, v_i\rangle$, если ее там не было.

14.32. Дан ориентированный граф. Найти все простые цепи в графе из вершины a в вершину b.

14.33. Дан ориентированный граф. Найти самый большой цикл в графе, то есть цикл, содержащий наибольшее количество дуг.

14.34. Ориентированный граф, не имеющий циклов, называется *деревом.* Проверить, является ли данный орграф деревом.

14.35. *Треугольником* в неориентированном графе называется тройка попарно соединенных дугами вершин. *Склеиванием треугольника* называется операция замены вершин треугольника новой вершиной с сохранением всех связей с остальными вершинами графа. Дан неориентированный граф. Склеить все треугольники графа.

14.36. Граф называется *полугамильтоновым*, если существует простая цепь, проходящая через все вершины графа. Дан орграф. Проверить, является ли он полугамильтоновым.

14.37. Граф называется *гамильтоновым*, если существует цикл, проходящий через все вершины графа. Дан орграф. Проверить, является ли он гамильтоновым.

14.38. Граф называется *полуэйлеровым*, если существует цепь, содержащая все дуги графа. Дан орграф. Проверить, является ли он полуэйлеровым.

14.39. Граф называется *эйлеровым*, если существует замкнутая цепь, содержащая все дуги графа. Дан орграф. Проверить, является ли он эйлеровым.

14.40. Граф называется *связным*, если любые две его вершины связаны простой цепью. Дан орграф. Определить, является ли он связным?

14.41. *Компонентой связности* графа назовем такое подмножество K его вершин, что любые две вершины K связаны простой цепью, но не существует цепи от любой вершины K до любой вершины, не принадлежащей K. Таким образом, граф может быть разбит на непересекающиеся компоненты связности. Дан орграф. Найти все его компоненты связности.

14.42. Решить задачи 12.26 – 12.27 с использованием динамических графовых структур.

15. Обработка текстов

А

15.1. Дан русский текст. Найти:
- а) частоты вхождений русских букв;
- б) количество вхождений слов различной длины;
- в) количество вхождений предлогов;
- г) наиболее часто встречающиеся в словах двух- и трехбуквенные последовательности;
- д) количество слов с шипящими согласными;
- е) количество слов со звонкими согласными.

15.2. Рассмотрим простейший способ шифровки текста – с помощью *перемешанного алфавита*. Каждую букву шифруемого текста заменяют на другую букву (ее шифр). Таблица соответствия букв и их шифров называется перемешанным алфавитом. Дан русский текст, перемешанный алфавит. Зашифровать и расшифровать текст с помощью перемешанного алфавита.

15.3. Метод шифровки текста с помощью *ключевого слова* состоит в следующем. Пронумеруем буквы русского алфавита (А–1, Б–2, В–3,. . . , Я–33), назовем эти номера «кодами» букв. Допустим, ключевое слово состоит из k букв ($k \geqslant 1$). Группа первых k букв исходного текста (пробелы, знаки препинания игнорируются) шифруется следующим образом: к коду первой буквы группы прибавить код первой буквы ключевого слова, если полученное значение больше 33, то вычесть 33. Полученное значение является кодом буквы, которой нужно зашифровать первую букву группы. Тот же прием нужно применить ко второй букве группы и второй группе ключевого слова и так далее до конца группы. Указанный алгоритм применить к следующим группам по k букв в каждой. Дан русский текст, ключевое слово. Зашифровать и расшифровать текст с помощью описанного алгоритма.

B

15.4. Исправление ошибок. Будем считать два слова близкими по написанию, если одно слово из другого можно получить за не более чем две операции, каждая из которых состоит в удалении или вставке буквы или замене одной буквы на другую. В первом входном файле задан текст, во втором – словарь правильно написанных слов. Написать программу, которая в диалоговом режиме предложит пользователю:

а) заменить каждое слово текста, для которого в словаре имеются близкие по написанию слова, на одно из таких слов;

б) включить в словарь отсутствующие там слова из текста.

Полученный в результате такой правки текст записать в выходной файл.

C

15.5. Задан текст, зашифрованный с помощью перемешанного алфавита (см. задачу 15.2). Заранее построив на основе реальных русских текстов таблицу частот букв алфавита, расшифровать текст. Программа расшифровки должна работать в диалоговом режиме, предлагая пользователю различные варианты расшифровки как для каждой буквы, так и для текста в целом, а пользователь имел бы возможность выбирать наиболее подходящий вариант.

15.6. Генератор сканвордов. Незаполненное поле сканворда задается в виде матрицы, состоящей из элементов трех видов: пустые клетки, в которые нужно при разгадывании записывать буквы; заштрихованные клетки; клетки-номера, в которых находятся номера загадываемых слов. Для удобства будем считать, что номера слов, расположенных по горизонтали, нечетные, а по вертикали – четные. Даны поле сканворда, база данных слов, в которой каждое русское слово сопоставлено с его определением. Сгенерировать сканворд, то есть для каждого номера указать определение слова так, чтобы сканворд имел корректное решение.

15.7. Написать программу, выводящую свой собственный текст. Ограничение: тривиальные программы, использующие открытие файла с исходным текстом или некоторые особенности конкретного языка (например, в Бейсике программа `10 LIST`), решением не считаются.

16. Игры

A

16.1. Дана шашечная позиция, то есть указаны координаты шашек на доске и их цвет. Указать все те шашки, которые находятся под боем шашек противника; необходимо учитывать, что ходы могут быть многошаговыми.

16.2. Решить задачу 16.1 для шахматной позиции.

16.3. Написать программу для игры двух человек в «крестики-нолики» на квадратной матрице порядка 3. Программа должна поочередно предоставлять соперникам право хода и выявлять победу одного из них или ничейную ситуацию.

16.4. Игра «пятнадцать». На квадратном поле 4×4 случайным образом расставлены 15 квадратных фишек с номерами от 1 до 15. Передвигая их поочередно в свободную позицию, игрок пытается расставить фишки в порядке возрастания номеров, начиная с левого верхнего угла. Написать программу, позволяющую играть в игру «пятнадцать» человеку.

B

16.5. Решить задачу 16.3 для игры человека и компьютера.

16.6. Написать программу для игры двух человек в «крестики-нолики» на бесконечном клетчатом поле. Программа должна поочередно предоставлять соперникам право хода и выявлять победу одного из них. Победившим считается тот игрок, который разместил пять своих значков подряд по вертикали, горизонтали или диагонали.

16.7. Решить задачу 16.4 для игры-соревнования человека с компьютером. Для человека и компьютера предоставляется по одному игровому полю со случайным образом расставленными фишками. Выигрывает тот, кто нашел решение за меньшее число шагов.

16.8. Игра «Нечестный морской бой». Правила игры: на клетчатом поле 10×10 каждого из двух игроков расставляется четыре однопалубных корабля (1 клетка), три двухпалубных (2 соседних клетки), два трехпалубных (три последовательных клетки) и один четырехпалубный (четыре последовательных клетки). Корабли не могут касаться «бортами» друг друга (не могут находиться на соседних клетках). Игроки делают поочередно ходы-«выстрелы» и сообщают их результаты: «мимо», «ранен», «потопил». В случае ранения или потопления ход со-

храняется у стрелявшего.
Написать программу, которая играет за одного из игроков, причем допускается «жульничество»: программа может иногда «подглядывать» за полем соперника.

16.9. Жизнь. Колония живых клеток развивается на бесконечном клетчатом поле. Развитие происходит периодами по следующим правилам: клетка выживает и переходит в следующий период, если она имеет двух или трех соседей из восьми возможных; если у клетки один сосед или нет соседей вовсе, то она погибает от изоляции; если соседей больше трех, то она погибает от перенаселения; в пустой позиции, у которой три живых соседа, рождается новая клетка. Написать программу, моделирующую развитие колонии клеток. Подобрать и исследовать такие начальные состояния колонии, при которых она бесконечно увеличивается; со всеменем погибает; циклически повторяет свое развитие.

16.10. Игра «Кто хочет стать миллионером?». Написать программу, моделирующую телевизионную игру «Кто хочет стать миллионером?». Программа задает случайную последовательность из 15 вопросов нарастающей сложности и предлагает игроку четыре варианта ответа на каждый вопрос, среди которых обязательно один правильный. Ценность вопросов определяется их номером (см. телеверсию игры). После неправильного ответа игра прекращается, деньги сгорают (кроме «несгораемой суммы», получаемой после пятого и десятого ответов).

16.11. Написать программу, обучающую устному счету. Программа должна предлагать человеку случайные задания и получать ответы. В случае неверного ответа может быть дана подсказка и предложены аналогичные задания с другими значениями. Систему оценки ответов разработать самостоятельно.

16.12. Написать программу, обучающую запоминанию исторических дат. Программа должна предлагать человеку случайные задания и получать ответы. В случае неверного ответа может быть дана подсказка и в дальнейшем вопрос должен быть повторен. Правила оценки ответов разработать самостоятельно.

16.13. Решить задачу 16.12 для обучения переводу чисел из одной системы счисления в другую.

16.14. Решить задачу 16.12 для заучивания иностранных слов и их переводов.

C

16.15. Решить задачу 16.6 для игры человека и компьютера.

16.16. Игра «Честный морской бой». Решить задачу 16.8 для случая, когда «жульничество» со стороны программы не допускается.

16.17. Игра «Морской бой с полным обманом» . Решить задачу 16.8 для случая, когда кроме указанного вида «жульничества» со стороны программы допускается во время игры перестановка своих кораблей в позиции, по которым еще не было выстрелов.

16.18. Игра «Ним». В игре участвуют три игрока. Даны три кучки камешков. Ход игрока состоит в том, что он забирает из какой-то одной (произвольной) кучки ненулевое число камешков. Игроки ходят по очереди. Выигрывает тот, кто заберет последний камешек. Написать программу, моделирующую игру «Ним», в которой участвуют два человека, а за третьего играет программа.

17. Задачи на разные темы

17.1. Обыкновенные дроби. Дано натуральное n. Вывести все различные обыкновенные несократимые дроби, принадлежащие интервалу от 0 до 1, знаменатели которых не превышают n.

17.2. Сколько, сколько? Дано натуральное $n(n \leqslant 9)$. Найти количество натуральных чисел, в десятичной записи которых содержится n значащих цифр и все эти цифры различны.

17.3. Сумма двух квадратов. Даны натуральные числа a и $b(a \leqslant b \leqslant 10^6)$. Найти количество натуральных чисел из отрезка $[a; b]$ таких, которые можно представить в виде суммы квадратов двух натуральных чисел по крайней мере двумя различными способами (две суммы, состоящие из одинаковых слагаемых, пусть даже с разным их порядком, считаются одинаковыми).

Пример.

a=*1*
b=*100*
Ответ=3

17.4. Сумма квадратов. Дано натуральное число a. Определить, можно ли его представить в виде суммы квадратов натуральных чисел: $a = {a_1}^2 + {a_2}^2 + \dots + {a_n}^2$ так, чтобы слагаемые этой суммы не повторялись; порядок слагаемых не важен. Если можно, то выдать найденную сумму; в случае, если таких способов несколько, выдать сумму с минимальным числом слагаемых.

17.5. Двумя способами. Дано натуральное число n. Найти наименьшее число s, которое может быть представлено в виде суммы $s = a^n + b^n$ (a и b натуральные) по крайней мере двумя нетривиально различными способами.

17.6. Барабан. Даны n чисел $a_1, a_2, \dots, a_n$, записанные по окружности. Вектором $x_k (1 \leqslant k \leqslant n)$ будем называть последовательность $a_k, a_{k+1}, \dots, a_n, a_1, \dots, a_{k-1}$. Будем считать, что вектор x_k меньше вектора x_m, если в первой неравной паре соответствующих элементов этих векторов выполняется условие $a_{k+i} < a_{m+i} (i = 0, 1, \dots)$. Среди векторов $x_1, x_2, \dots, x_n$ найти минимальный.

17.7. Конечная последовательность $a_1, a_2, \dots, a_n$ состоит из цифр 0, 1 и 2. Участком последовательности назовём любые k подряд идущих членов последовательности $(1 \leqslant k \leqslant n)$. Два участка называются смежными, если сразу после последнего члена одного участка идет первый член другого участка.
Дано натуральное $n (2 \leqslant n \leqslant 1000)$. Найти какую-нибудь последовательность цифр 0, 1 и 2, чтобы в ней не было одинаковых смежных участков.

17.8. Такая функция. Определим функцию $f(x)$ для целого неотрицательного аргумента x: $f(0) = 0, f(1) = 1, f(2n) = n, f(2n + 1) = f(n) + f(n + 1)$. Дано $N (0 \leqslant N \leqslant 10^9)$. Найти $f(N)$. Ограничение: рекурсией и массивами не пользоваться.

17.9. Точки на прямой. Точки на числовой прямой пронумерованы числами $1, 2, 3, \dots, n (2 \leqslant n \leqslant 1000)$. Известны расстояния между некоторым из этих точек: действительное $a_{i,j}$ равно расстоянию между точками с номерами i и j. Проверить, можно ли расположить эти точки на числовой прямой так, чтобы расстояния между ними соответствовали заданным.

Техническое требование. В первой строке входного файла указано количество точек n. В остальных строках находятся тройки чисел: номер первой точки, номер второй точки, расстояние между ними. Номера точек — натуральные числа, не превосходящие n; расстояние задается вещественным неотрицательным числом не больше 1000. Программа выдает в выходной файл слово "да", если точки можно расположить на числовой прямой, в противном случае выдается слово "нет".

17.10. Точки на плоскости. Имеется n точек плоскости ($2 \leqslant n \leqslant 100$) и известны расстояния между некоторыми из них. Написать программу, которая выдает на экран слово "да", если эти точки можно расположить на плоскости, или "нет" — в противном случае.

Техническое требование. В первой строке входного файла указано количество точек n. В остальных строках находятся тройки чисел: номер первой точки, номер второй точки, расстояние между ними. Номера точек — натуральные числа, не превосходящие n; расстояние задается вещественным неотрицательным числом не больше 1000. Ответ задачи программа выдает в выходной файл.

17.11. Системы счисления. Множество цифр системы счисления с основанием $R (2 \leqslant R \leqslant 36)$ содержит цифры $0, 1, \ldots 9$; если количества цифр до R не хватает, то в множество добавляются прописные буквы латинского алфавита $A, B, \ldots$. Даны две строки символов. Если эти строки могут являться записями некоторого числа в системах счисления с основаниями r_1 и r_2 соответственно, то найти минимальные значения r_1 и r_2; если таких оснований не существует, то выдать число 0.

17.12. Обращение. Обращением натурального числа n назовём такое натуральное число $n_{\text{обр}}$, что запись числа $n_{\text{обр}}$ в двоичной системе счисления получается переворотом двоичной записи числа n. Например, для $n = 22_{(10)} = 10110_{(2)}$ число $n_{\text{обр}}$ равно $13_{(10)} = 1101_{(2)}$.

Даны два натуральных числа a и b ($a \leqslant b \leqslant 10^9$). Найти наименьшее и наибольшее из обращений натуральных чисел отрезка $[a; b]$.

17.13. Достроить забор. Матрица размером $m \times n$ заполнена нулями и единицами. Забором будем называть последовательность единичных элементов матрицы, таких, что два последовательных элемента забора являются соседями по горизонтали, вертикали или диагонали. Таким образом, заборы вместе с краями таблицы ограничивают друг от друга области, заполненные нулями (зоны).

Даны координаты двух нулевых элементов матрицы (заключенные) $(x_1; y_1), (x_2; y_2)$. Если они принадлежат разным зонам, то выдать число 0; если заключенные находятся в одной зоне, то выдать число, равное наименьшему количеству нулевых элементов матрицы, при замене которых на единицы заключенные будут находиться в разных зонах.

17.14. Прямые. Даны натуральное число n, действительные числа $a_1, b_1, c_1, a_2, b_2, c_2, \ldots, a_n, b_n, c_n$. Найти количество областей плоскости, на которые её разбивают прямые, задаваемые уравнениями $a_i x + b_i y + c_i = 0, i = 1, 2, \ldots, n$.

17.15. Домино. Имеется n костей домино, возможно, некоторые кости одинаковы. Найти длину самой длинной цепочки костей, выстроенной по правилам домино.

Техническое требование. В первой строке входного файла указано количество костей n $(1 \leqslant n \leqslant 100)$. В каждой из следующих n строк находится пара целых чисел, разделенных пробелом: значения очередной кости a_i, b_i, где $0 \leqslant a_i, b_i \leqslant 6$. В выходной файл нужно поместить одно натуральное число, равное длине наидлиннейшей цепочки костей в соответствии с условиями задачи.

17.16. Удалить числа. Дана последовательность целых чисел $a_1, a_2, \ldots, a_n$. Найти минимальное количество чисел, после удаления которых из последовательности оставшиеся числа образуют возрастающую последовательность.

17.17. Найти прогрессию. Дано множество целых чисел a_1, a_2, $\ldots$, a_n. Построить из элементов множества наиболее длинную последовательность, которая является:

а) арифметической прогрессией;

б) геометрической прогрессией.

17.18. Тождественность выражений. Заданы записи двух арифметических выражений с целыми числами, операциями сложения, вычитания, умножения, деления нацело, круглыми скобками и переменными. Написать программу, которая проверяет тождественность выражений, то есть совпадают ли значения выражений для всех возможных значений переменных.

17.19. Тавтология. Дана запись логического выражения в соответствии с синтаксисом:

```
<лвыраж> ::= <лтерм>|<лтерм><оп><лвыраж>
<лтерм> ::= <лприм>|<лприм> & <лтерм>
<оп>::=v|x
<лприм> ::= T|F|<имя>|(<лвыраж>)
```

где <имя> – правильно построенный идентификатор. Проверить, не является ли данное выражение тавтологией, когда для всех возможных логических значений переменных значение выражения всегда истинно.

17.20. Оптимальное размещение. На клетчатом поле задано несколько фигур. Каждая фигура представляет собой множество смежных клеток, причем каждая клетка фигуры соприкасается, по крайней мере, одной из своих сторон с одной из соседних клеток этой же фигуры. Фигуры друг с другом не пересекаются (то есть у них нет общих клеток). Требуется указать размеры минимального по площади прямоугольника, в котором можно разместить заданные фигуры без пересечений. Разрешается их поворачивать на угол, кратный $\frac{\pi}{2}$.

Техническое требование. Структура входного файла input.txt следующая:

```
<РазмерПоляN><РазмерПоляM>
<a11><a12>...<a1M>
...
```

```
<aN1><aN2>...<aNM>
```

где aij — номер фигуры (натуральное число), которой принадлежит клетка поля с координатами i и j, либо ноль, если клетка не принадлежит ни одной фигуре. В выходной текстовый файл output.txt программа записывает одно число — площадь искомого прямоугольника (количество клеток в нём).

Пример.

Входной файл:

```
4 7
0 0 1 0 0 0 0
1 0 1 2 2 2 2
1 0 1 0 0 0 2
1 1 1 0 2 2 2
```

Выходной файл: 16

17.21. Подматрицы. Дана вещественная прямоугольная матрица порядка $M \times N$. Подматрицей назовем совокупность смежных элементов матрицы a_{ij}, индексы которых принадлежат отрезкам: $i \in [m_1, m_2]$, $j \in [n_1, n_2]$, где $1 \leqslant m_1 \leqslant m_2 \leqslant M$ и $1 \leqslant n_1 \leqslant n_2 \leqslant N$. Найти подматрицу данной матрицы с наибольшей суммой элементов.

17.22. Факториальное представление. Факториальным представлением целого числа N назовем набор целых значений $(d_n, d_{n-1}, \ldots, d_2, d_1)$, для которого

$$N = d_n n! + d_{n-1}(n-1)! + \cdots + d_2 2! + d_1 1!$$

Дано факториальное представление целого числа N. Найти N.

Пример.

Вход: *-1 0 2 1*

Выход: -19

17.23. Найти факториальное представление. Решить задачу, обратную 17.22. Дано целое число N. Требуется найти одно из его факториальных представлений в соответствии с одним из следующих ограничений или сообщить, что это невозможно:

а) количество ненулевых коэффициентов в представлении должно быть не менее данного натурального k;

б) $|d_i| \leqslant a$, где $1 \leqslant i \leqslant n$, $a > 0$ — данное целое число.

17.24. Факториальная система счисления. Представлением натурального числа N в факториальной системе счисления называется набор целых значений $(d_n, d_{n-1}, \ldots, d_2, d_1)$, для которого

$$N = d_n n! + d_{n-1}(n-1)! + \cdots + d_2 2! + d_1 1!, \text{ где } d_i \leqslant i,\ i = 1, 2, \ldots, n.$$

Дано натуральное число N. Найти его представление в факториальной системе счисления.

17.25. Фибоначчиева система счисления. Записью натурального числа N в фибоначчиевой системе счисления (ФСС) называется последовательность двоичных цифр $a_n a_{n-1} \ldots a_{2(\Phi)}$ ($a_i = 0$ или $a_i = 1$), такая, что

$$N = a_n f_n + a_{n-1} f_{n-1} + \cdots + a_2 f_2,$$

где $f_2, \ldots, f_{n-1}, f_n$ — числа Фибоначчи ($f_1 = f_2 = 1, f_i = f_{i-1} + f_{i-2}, i \geqslant 3$), а f_n — наибольшее из не превосходящих N чисел Фибоначчи. Например, $30_{(10)} = 1010001_{(\Phi)}$, так как $30 = 1 \cdot 21 + 0 \cdot 13 + 1 \cdot 8 + 0 \cdot 5 + 0 \cdot 3 + 0 \cdot 2 + 1 \cdot 1$. Дано натуральное N. Найти его запись в фибоначчиевой системе счисления.

17.26. Снова ФСС. Не всякая последовательность нулей и единиц может являться записью натурального числа в фибоначчиевой системе счисления (см. задачу 17.25) Например, последовательность 110 не является записью числа в ФСС, так как число $5 = 1 \cdot 3 + 1 \cdot 2 + 0 \cdot 1$ записывается иначе: $5 = 1 \cdot 5 + 0 \cdot 3 + 0 \cdot 2 + 0 \cdot 1$, то есть $5 = 1000_{(\Phi)}$ (смотрите интересную книгу о числах Фибоначчи [15]). Дано натуральное M. Найти количество правильных фибоначчиевых записей натуральных чисел среди всевозможных последовательностей, состоящих из M двоичных цифр (нулей и единиц). Ведущие незначащие нули можно отбрасывать.

17.27. Задача о прыгуне. Прыгун может прыгать в одном направлении по разделенной на клетки полосе. За каждый прыжок он может переместиться либо в соседнюю клетку, либо через одну клетку. Изначально прыгун находится на клетке с номером 1. Сколькими различными способами он может допрыгать до клетки с номером n? Способы прыгания считаются одинаковыми, если в ходе каждого из них прыгун побывал в одних и тех же клетках.

17.28. Тройной прыжок разрешен! Решить задачу 17.27 в предположении, что за каждый прыжок прыгун может переместиться в соседнюю клетку либо через одну клетку, либо через две клетки.

17.29. Двоичный палиндром. Целое число называется двоичным палиндромом, если его запись в двоичной системе счисления (без незначащих нулей) является палиндромом. Даны целые числа N и M ($0 \leqslant N \leqslant M \leqslant 50000$), записанные в десятичной системе счисления. Найти количество двоичных палиндромов, принадлежащих отрезку $[N; M]$.

17.30. Китайская теорема об остатках. Даны натуральные взаимно простые числа $m_1, m_2, \ldots, m_n$, целые неотрицательные числа $r_1, r_2, \ldots, r_n (m_i > r_i$ для каждого $i \in [1; n])$. Согласно китайской теореме об остатках, существует единственное число X, такое, что $X < m_1 \cdot m_2 \cdot \ldots \cdot m_n$ и для каждого целого $i \in [1; n]$ остаток от деления X на m_i равен r_i. Требуется по наборам чисел $\{m_i\}$ и $\{r_i\}$ найти число X.

17.31. Игра с суммой. Квадратная таблица, в ячейках которой располагаются целые числа, имеет размеры $N \times N$, где N–

нечетно. В центральной клетке таблицы стоит число 0, сумма всех чисел равна нулю. В центре находится фишка. Играют два игрока, поочередно делая ходы. За один ход игрок может переместить фишку из ячейки в любую другую соседнюю ячейку, в которую еще не было сделано ранее ходов (соседней ячейкой назвается любая из не более чем восьми смежных ячеек). После хода в ячейку она закрывается, и в неё больше ходить нельзя. При каждом ходе вычисляется сумма игры s, то есть число ячейки прибавляется к s. Игра заканчивается, когда нельзя сделать ни одного хода; ходы игрокам пропускать нельзя. Если в результате игры сумма s получилась отрицательной, то первый игрок выплачивает второму $-s$ рублей, если положительной, то второй игрок первому выплачивает s рублей.
Написать программу, играющую за одного из игроков по данной таблице чисел.

17.32. Даны натуральные n и m $(m \leqslant n)$. Получить все последовательности, состоящие из n двоичных цифр (нулей и единиц), содержащие ровно m единиц.

17.33. Даны вещественные числа $a_1, a_2, \ldots, a_n (a_i \neq 0, 1 \leqslant i \leqslant n)$. Найти набор индексов $i_1, i_2, \ldots, i_k (1 \leqslant i_1 \leqslant i_2 \leqslant \cdots \leqslant i_k \leqslant n)$ такой, чтобы сумма $a_{i_1} + a_{i_2} + \cdots + a_{i_k}$ принимала бы наименьшее по модулю значение из всех подобных сумм.

17.34. Требуется налить. Даны n сосудов, имеющих объемы $v_1, v_2, \ldots, v_n$, где v_i — целые числа, обозначающие литры. Ответить на вопрос, можно ли с помощью указанных сосудов отмерить ровно v литров жидкости?

17.35. НОД чисел Фибоначчи. Даны натуральные m и n $(m \leqslant n \leqslant 2^{15})$. Найти число различных пар чисел Фибоначчи (f_i, f_j), где $m \leqslant i \leqslant j \leqslant n$ таких, чтобы их наибольший общий делитель тоже был бы числом Фибоначчи.

17.36. Требуется нарезать. Дана непустая последовательность, состоящая из двоичных цифр (длина последовательности $l \leqslant 255$), и дано натуральное число $n (n \leqslant l)$. Разбить последовательность на n непустых групп последовательных цифр так, чтобы сумма чисел, двоичная запись которых состоит из последовательности цифр групп, была бы

а) минимальной;

б) максимальной.

17.37. Посыпались. На экран компьютера выведено n-значное натуральное число, записанное в десятичной системе счисления. В результате действия вируса некоторые цифры числа «посыпались», то есть выпали из числа, а вместо них остались пустые знакоместа. Сколько различных значений можно получить, вставляя выпавшие цифры обратно в число? Вставлять в одно знакоместо можно только одну цифру.

Техническое требование. На входе две строки: в первой задан шаблон числа, в котором место каждой выпавшей цифры обозначено знаком "?", длина шаблона не превосходит 12; во второй строке дан список выпавших цифр (без пробелов или других знаков), причем количество знаков "?" в шаблоне совпадает с количеством цифр во второй строке. Выдать целое число, равное количеству различных значений, которые возможно получить вставкой выпавших цифр в число вместо знаков "?".

17.38. Упражнение Р.У. Хемминга. Все натуральные числа, которые можно представить в виде $2^i 3^j 5^k$, где i, j и k - целые неотрицательные, выстроим в возрастающую последовательность $1, 2, 3, 4, 5, 6, 8, 9, 10, 12 \ldots$ Дано N — одно из чисел последовательности. Найти следующее после N число последовательности.

17.39. Даны три натуральных числа a, b, c $(1 \leqslant a \leqslant b \leqslant 10^9, 1 \leqslant c \leqslant 10)$. Найти количество всех чисел z, принадлежащих отрезку $[a; b]$, таких, что они могут быть представлены в виде $z = z_n c^n + z_{n-1} c^{n-1} + \cdots + z_1 c + z_0$, где $z_i \in \{0, 1\}, n \geqslant 0$.

17.40. Перевод в двоичную систему. Дано натуральное N, записанное в шестнадцатеричной системе счисления ($1 \leqslant N \leqslant 2^{2004} - 1$). Найти количество единиц в двоичной записи этого числа.

17.41. Покрытие прямоугольника. Дан прямоугольник с размерами $a \times b$ и n кругов с радиусами $r_1, r_2, \ldots r_n$. Какое наименьшее количество из данных кругов надо взять, чтобы ими полностью покрыть прямоугольник?

17.42. Квадраты. На координатной плоскости заданы n квадратов, стороны которых параллельны осям. Требуется найти площадь области, покрытой квадратами.

Техническое требование. . В первой строке входного файла заданы находится целое число $n (0 \leqslant n \leqslant 100)$ — количество квадратов. В каждой из n следующих строк находится тройка чисел x, y, a, где x, y— координаты левого нижнего угла квадрата, a— длина его стороны; все числа целые, $-30000 \leqslant x, y \leqslant 30000$, $0 \leqslant a \leqslant 1000$. В выходной файл необходимо записать площадь области объединения квадратов в виде вещественного числа с точностью до двух знаков после запятой.

17.43. Одна строка. Дан текстовый файл, состоящий из n строк ($1 \leqslant n \leqslant 50000$), длина каждой строки не превосходит 255 символов. В файле есть одна строка, которая встречается только один раз, все остальные строки повторяются, причем обязательно четное количество раз. Найти эту уникальную строку.

17.44. Еще раз одна строка. Решить задачу 17.43, отменив ограничение четности вхождений повторяющихся строк, то есть только одна строка появляется единожды, другие повторяются два или более раз.

17.45. Даны натуральные m, n $(2 \leqslant m \leqslant 16, 3 \leqslant m + n \leqslant 24)$. Найти количество n-значных натуральных чисел в системе счисления с основанием m, таких, что их запись в этой системе счисления не содержит двух подряд значащих нулей.

17.46. Метро. В городе есть метрополитен, состоящий из n станций. Станции соединены друг с другом лининями, причём две станции могут быть соединены только одной линией. В исходном состоянии система метро является связной, то есть с любой станции до любой другой можно добраться, хотя бы и через другие станции.

В связи с убыточностью принято решение метро закрыть, причем закрывать метро будут постепенно, в год по одной станции. Естественно, после закрытия какой-то станции все линии, к ней подходящие от других станций, тоже прекращают действовать. Требуется указать такой порядок закрытия станций, чтобы после каждого закрытия оставшаяся часть метрополитена оставалась связной.

Техническое требование. В первой строке входного файла находятся два целых неотрицательных числа n и m — соответственно количество станций и количество линий между станциями $1 \leqslant n \leqslant 100, 0 \leqslant m < 10000$. В следующих m строках находится информация о линиях между станциями, каждая линия обозначена двумя натуральными числами b_i и e_i, которые равны номерам соединяемых этой линией станций. Считать, что система станций и линий задана корректно, то есть не может быть двух линий между одинаковыми станциями, номера станций не превосходят n, система связна. В выходном файле в n строках разместить номера станций в том порядке, в котором их нужно закрывать по условию задачи (в каждой строке – один номер).

17.47. По парам. Дано натуральное число $N (N \leqslant 10^5)$. Найти максимальное количество пар натуральных чисел, не превосходящих N, таких, чтобы сумма чисел пары была бы простым числом, причем каждое число может входить лишь в одну пару.

Пример.

Введите N=*8*

Ответ: количество пар=4

17.48. Дано натуральное число a ($1 \leqslant a \leqslant 10^9$). Найти минимальное натуральное число n, чтобы n^n делилось бы на a.

17.49. Сумма. Даны натуральные числа $a_1, a_2, \ldots, a_n$. Найти минимальное число, которое нельзя представить в виде суммы некоторых из данных натуральных чисел (каждое из a_i может входить в сумму только один раз; сумма может состоять и из одного слагаемого).

17.50. Кони, кони... Найти все возможные расстановки двенадцати коней на шахматной доске, при которых каждое поле доски было бы под ударом хотя бы одного из них.

17.51. Миролюбивые магараджи. Будем считать, что «шахматная» фигура «магараджа» может ходить как ферзь и как конь. Расставить на доске 10×10 десять магараджей так, чтобы они не угрожали друг другу.

17.52. Многопроцессорность. В конфигурацию компьютера включены n одинаковых процессоров ($n \geqslant 1$). На вход последовательно поступают задания; каждое из заданий может быть направлено для выполнения на любой из свободных процессоров, причем оно занимает этот процессор до своего полного завершения. Прерываний исполнения процессором заданий не происходит. Очередное поступившее задание направляется на любой из освободившихся процессоров.
На компьютере надо выполнить m заданий, которые требуют времени $t_1, t_2, \ldots, t_m$ соответственно ($t_i > 0, i = 1, 2, \ldots, m$). Определить такую очередность поступления заданий, чтобы задания были выполнены за минимальное время.

17.53. Разложить факториал. Дано натуральное число $n (n \leqslant 1000)$. Разложить число $n!$ в произведение простых чисел (если простое число входит в разложение несколько раз, то выдать его один раз и указать показатель степени – количество вхождений множителя).

17.54. Делители факториала. Даны натуральные числа $n (n \leqslant 1000)$, a и b $(a \leqslant b \leqslant 10^6)$. Найти количество делителей числа $n!$, принадлежащих отрезку $[a, b]$.

17.55. Игра с лунками. Имеется n лунок, расположенных в линию $(n \geqslant 3)$. В каждой лунке может находиться не более чем по одному шару. В первых k лунках находятся белые шары $(1 \leqslant k \leqslant n-2)$, $(k+1)$-я лунка пустая, в остальных $n-k-1$ лунках лежат черные шары. За один ход можно шар из любой лунки переместить либо в соседнюю пустую лунку, либо через одну лунку в пустую. За какое минимальное количество ходов можно все белые шары переместить в конец ряда лунок, а все черные — в его начало?

17.56. Многоугольник. Многоугольник на плоскости задается координатами своих вершин $x_1, y_1, x_2, y_2, \ldots, x_n, y_n (n \leqslant 1000)$. Найти количество точек с целочисленными координатами, лежащих внутри многоугольника, но не на его границе. Считать, что последовательность точек такова, что несмежные стороны многоугольника не пересекаются.

17.57. Конфуз. Пусть a - целочисленный массив, состоящий из n элементов $(2 \leqslant n \leqslant 10^4)$ $a_1, a_2, \ldots, a_n$, где $|a_i| \leqslant 2 \cdot 10^9, i = 1, 2, \ldots, n$. Вычислим сумму элементов S, $S = a_1 + a_2 + \cdots + a_n$. Заменим каждый элемент массива на разницу S и этого элемента: $a_i := S - a_i, 1 \leqslant i \leqslant n$. Такое преобразование массива a назовем операцией Confuse. Напишите программу, которая по массиву b, полученному в результате k-кратного применения операции Confuse к некоторому массиву a $(1 \leqslant k \leqslant 100)$, вычислит разность максимального и минимального элементов исходного массива a.

17.58. Абракадабра. Последовательность строк $s_0, s_1, s_2, s_3 \ldots$ строится следующим образом: s_0— пустая строка, а каждая по-

следующая строка получается удваиванием предыдущей строки и приписыванием к ней слева очередной буквы латинского алфавита (начиная с буквы a). Таким образом,
$s_0 =$""; $s_1 =$"a"; $s_2 =$"baa"; $s_3 =$"cbaabaa"; ...
Даны натуральные числа n, m $(1 \leqslant n \leqslant 26; 1 \leqslant m \leqslant 2^n - 1)$. Определить символ, который стоит в m-й позиции в строке s_n.

17.59. Гвозди. На прямоугольном столе разложили n одинаковых листков бумаги со сторонами, параллельными краям стола. Требуется прибить листки к столу гвоздями, причем так, чтобы каждый гвоздь прикреплял бы к столу хотя бы один листок и каждый листок был бы прибит ровно одним гвоздём. Гвозди нельзя забивать в границы листков. Считать, что длина каждого листка параллельна оси Ox, а ширина параллельна оси Oy.

Техническое требование. В первой строке входного файла записаны три числа: n– количество листков бумаги $(1 \leqslant n \leqslant 100)$, a и b размеры каждого листка $(1 \leqslant a, b \leqslant 1000)$. В каждой из следующих n строк указано по паре чисел x_i, y_i– координаты левого нижнего угла очередного листка $(0 \leqslant x_i, y_i \leqslant 1000)$. Все числа целые.
В выходной файл нужно поместить целое число, равное 0, если такого способа прибить гвозди не существует, или равное минимальному количеству гвоздей, которыми прибивается бумага в соответствии с условием задачи.

17.60. Большое число. Даны натуральные числа $m, n (1 \leqslant m, n \leqslant 10^{100})$. Найти последнюю цифру числа m^n.

17.61. Вот такие фишки. На прямоугольном клетчатом поле размером $m \times n$, где m обозначает количество горизонтальных рядов, а n— количество вертикалей, размещены n фишек, по одной в каждой вертикали; каждая фишка занимает одну клетку. За один ход можно переместить любую фишку на одну клетку вверх или вниз по вертикальному ряду клеток. Найти минимальное количество ходов, за которые все фишки можно выстроить в одну (любую) горизонталь.

Техническое требование. В первой строке входного файла указаны два натуральных числа m, n $(1 \leqslant m, n \leqslant 10^4)$. Далее в файле размещены n натуральных чисел, каждое i-е число принадлежит отрезку $[1; m]$ и обозначает, в какой горизонтали изначально находится i-я фишка. В выходной файл необходимо записать одно натуральное число, обозначающее минимально необходимое количество ходов, за которые все фишки можно выстроить в один ряд.

17.62. Шарики. В двух ящиках находятся шарики, в первом m штук, во втором n штук $(1 \leqslant m + n \leqslant 2 \cdot 10^9)$. За один ход можно из одного ящика переложить в другой ящик столько шариков, сколько в нём уже находится. Можно ли переложить все шарики в один из ящиков?

17.63. Компьютерная сеть. В некоторой организации решили создать компьютерную сеть. Для этого от единого сервера к каждому из компьютеров сети нужно проложить отдельный кабель. Но руководители организации решили сэкономить и не закупать новый кабель, а использовать для этого обрезки старого кабеля. Два обрезка кабеля можно спаять, получив один более длинный обрезок. Любой обрезок можно разрезать на два. Надо определить, за какое минимальное количество спаек можно сделать нужное количество соединений компьютеров с сервером. Разрешается делать неограниченное количество разрезов обрезков кабеля.

Техническое требование. В первой строке входного файла даны два числа: m— количество подключаемых компьютеров к серверу, n— количество имеющихся обезков кабеля $(1 \leqslant m \leqslant 100, 1 \leqslant n \leqslant 1000)$. В следущих m строках находятся числа, по одному в каждой строке. Они обозначают расстояния от каждого из m компьютеров до сервера. В следующих n строках расположены числа, по одному в каждой строке, которые обозначают длины имеющихся обрезков кабеля. Все числа целые положительные. Программа в выходной файл должна выдавать ответ в виде целого числа, равного -1, если создать сеть в соответствии с указанными условиями нельзя, или равного минимальному количеству спаек обрезков, сделав которые, можно соединить каждый компьютер с сервером.

Примечание. Надеемся, что в реальных организациях руководство не будет принимать подобных решений о создании сети на основе таких паяных-перепаяных кабелей ...

17.64. Упорядочим числа по алфавиту. Даны натуральные n, k, m $(1 \leqslant n \leqslant 10^6; 2 \leqslant k \leqslant 36; 1 \leqslant m \leqslant n$. Расмотрим последовательность десятичных записей первых n натуральных чисел. Например, для $n = 10$ получим:

$1_{(10)}, 2_{(10)}, 3_{(10)}, 4_{(10)}, 5_{(10)}, 6_{(10)}, 7_{(10)}, 8_{(10)}, 9_{(10)}, 10_{(10)}$.

Запишем эти числа в системе счисления с основанием k, равного, например, 3:

$1_{(3)}, 2_{(3)}, 10_{(3)}, 11_{(3)}, 12_{(3)}, 20_{(3)}, 21_{(3)}, 22_{(3)}, 100_{(3)}, 101_{(3)}$.

Упорядочим записи этих чисел лексикографически, то есть так, как принято в словарях:

$1_{(3)}, 10_{(3)}, 100_{(3)}, 101_{(3)}, 11_{(3)}, 12_{(3)}, 2_{(3)}, 20_{(3)}, 21_{(3)}, 22_{(3)}$.

Переведем опять в десятичную систему:

$1_{(10)}, 3_{(10)}, 9_{(10)}, 10_{(10)}, 4_{(10)}, 5_{(10)}, 2_{(10)}, 6_{(10)}, 7_{(10)}, 8_{(10)}$.

Требуется найти, чему равно m-е число этой последовательности (то есть найти десятичную запись m-го числа последовательности натуральных чисел от 1 до n, упорядоченной лексикографически в соответствии с k-ичными записями этих чисел.).

Пример.

n=*10*
k=*3*
m=*5*
Ответ: это число 4

Примечание: считать, что буквы в алфавите идут после цифр.

17.65. Двоичное плюс-минус. Дано натуральное число n в двоичной системе счисления ($1 \leqslant n \leqslant 2^{100}$). Найти результат сложения его с 1 и вычитания из него 1.

17.66. Двоичное минус степень. Дано натуральное число n, записанное в двоичной системе счисления, и целое m ($1 \leqslant n \leqslant 2^{100}; 0 \leqslant m \leqslant 100$). Найти разность n и 2^m.

17.67. Уравнение в целых. Даны целые a_1, a_2, a_3, a_4, b. Решить уравнение $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = b$ в целых числах, то есть найти все четверки целочисленных значений $(x_1; x_2; x_3; x_4)$, обращающих уравнение в тождество.

17.68. Из a сделать b. Даны два целых положительных числа a, b. Найти минимальное количество цифр двоичной записи числа a, вычеркнув которые можно получить двоичную запись числа, равного b, или сообщить, что это сделать невозможно.

Пример.

a=*73*
b=*5*
Ответ: количество цифр равно 2

17.69. Произведение цифр. Дано натуральное число $n \leqslant 10^6$. Найти наименьшее натуральное число, произведение цифр десятичной записи которого равнялось бы n, или сообщить, что таких чисел не существует.

Пример.

Введите n=*120*
Ответ:358

17.70. Несократимые дроби. Даны натуральные числа n и d ($0 \leqslant n < d \leqslant 255$). Найти количество всех обыкновенных правильных несократимых дробей, имеющих различные значения, числитель которых не превосходит n, а знаменатель не превосходит d.

Пример.

Введите n=*3*
Введите d=*5*
Ответ: количество дробей равно 9

17.71. «Реверси». Игра «Реверси» имеет следующие правила. Квадратная доска разбита на $N \times N$ клеток ($3 \leqslant N \leqslant 10$). Играют двое, делая по очереди ходы. В игре используются особые фишки, с одной стороны фишка имеет белый цвет (цвет первого игорка), с другой– черный (цвет второго). Каждый игрок за один ход может поместить одну фишку в любую пустую клетку доски своим цветом вверх. Например, свой ход сделал игрок белыми фишками. Если после этого хода на доске по горизонтали, вертикали или диагонали складывается последовательность фишек черного цвета, ограниченная по краям двумя белыми фишками, то все эти черные фишки переворачиваются и становятся белыми; таких последовательностей может получиться несколько. Аналогично ходит игрок черными фишками. Игра заканчивается, когда вся доска заполнена фишками. Победившим считается игрок, у которого на доске больше фишек с его цветом; возможна ничья.
Напишите программу, моделирующую «Реверси» и играющую за одного из игроков.

17.72. Параллельные вычисления. Во входном файле задана программа, состоящая из N операторов присваивания $1 \leqslant N \leqslant 16$; синтаксис:

```
<присв> ::= <идент>:=<идент><оп><идент>
<оп>::=+|-|*|/
<идент> ::= A|B|...|Z
```

Требуется распределить выполнение этих операторов между двумя процессорами таким образом, чтобы общее время выполнения было минимальным и при этом результат выполнения был бы эквивалентен исходной программе. Будем считать, что для выполнения одного оператора процессору требуется один такт работы. Введем еще один оператор NULL, который ничего не изменяет, но процессор также тратит на него один такт. Очевидно, что два оператора присваивания не могут быть одновременно выполнены на разных процессорах, если в левой части одного из них содержится идентификатор переменной, которая есть в правой части другого.

Техническое требование. Во входном текстовом файле дан текст программы, в соответствии с вышеприведенным синтаксисом. В каждой строке расположен один оператор присваивания, пустых строк нет.

В выходной файл требуется поместить результат распределения программы на два процессора в следующем виде: в первой строке указывается целое положительное число T– время выполнения программы (в тактах работы). В каждой из T следующих строк располагаются по два оператора, выполняемых одновременно: первый– для первого процессора, второй– для второго. Операторы разделены, по крайней мере, одним пробелом. В качестве оператора может выступать присваивание либо оператор NULL.

Пример.

Во входном файле:

```
A:=B+C
B:=C-D
M:=K*N
E:=M/B
```

В выходном файле:

```
3
A:=B+C M:=K*N
B:=C-D NULL
NULL  E:=M/B
```

17.73. Нарушенный баланс. В тексте письма могут быть буквы, цифры, знаки препинания, круглые скобки и другие символы. Если текст составлен правильно, то в нем соблюдается баланс круглых скобок, то есть каждой открывающей скобке соответствует далее по тексту одна закрывающая, причем между ними может быть сбалансированная по скобкам часть текста. Текст (или его часть), не содержащий ни одного символа или не содержащий круглых скобок, считается сбалансированным по скобкам.

При передаче правильно составленного письма по электронной почте в результате сбоя из текста исчезло некоторое количество скобок. Необходимо определить, сколько существует различных способов вставки недостающего минимального числа скобок в поврежденный текст так, чтобы восстановить баланс скобок. Вставлять скобки можно между любыми символами текста, а также перед первым и после последнего символа.

Техническое требование.

Вход: входной текст читается из текстового файла с именем *INPUT.TXT*. Максимальная длина текста — 100 символов. Пробелы и символы окончания строк не считаются частью текста и должны быть проигнорированы.

Выход: на экран выводится целое число, равное

а) количеству различных способов вставки недостающего минимального числа скобок, если баланс скобок нарушен и его вставкой скобок можно восстановить, или

б) 0, если баланс скобок не нарушен, или

в) -1, если баланс скобок в тексте нельзя восстановить ни при какой вставке скобок.

Максимальное время работы программы — 15 секунд.

Пример.

```
x)a(;-)(x()(
```

Выход: 10

17.74. Дроби. Дано действительное число в десятичной записи. Необходимо указать равное ему смешанное число с выделенной целой частью, дробная часть которого является правильной несократимой обыкновенной дробью.

Техническое требование.

Вход: входной текст читается из текстового файла с именем INPUT.TXT. В файле задано действительное число (возможно, отрицательное; возможно, с дробной частью; возможно, с периодом) согласно синтаксису:

```
<Действительное> ::= [-]<Целая часть>[.<Дробная часть>]
<Целая часть>    ::= <Целое>
<Дробная часть>  ::= <Целое>(<Целое>)
                   | <Целое>
                   | (<Целое>)
```

Здесь <Целое> означает любую последовательность цифр 0, 1, 2, ..., 9. В круглых скобках указывается период числа. Максимальное количество цифр в целой части равно 9. Максимальное количество цифр в дробной части равно 9.

Выход: выдать на экран ответ в виде смешанного числа с целой частью и дробной частью. Дробная часть должна быть правильной несократимой обыкновенной дробью. Если или целая часть равна нулю, или дробная часть равна нулю, то ее выводить не нужно. Если число равно нулю, то вывести только целую часть числа. Целую часть от дробной части отделить одним пробелом, числитель дробной части от знаменателя отделить знаком "/".

Максимальное время работы программы — 15 секунд.

Пример.

Вход: *-1.2(34)*

Выход: -1 116/495

17.75. Решить задачу, обратную задаче 17.74: дана запись смешанного числа (в соответствии с правилами из задачи). Найти десятичную запись этого числа в виде, указанном в задаче 17.74.

17.76. «Сапер». Известная игра «Минер» устроена так:

минное поле представляет собой клетчатый прямоугольник, в клетках которого могут быть мины. Каждая клетка содержит либо мину, либо число мин в соседних клетках (у каждой клетки может быть до восьми соседей). Все клетки изначально скрыты. Каждый шаг игрока заключается в попытке вскрыть одну из клеток поля. Если там оказывается мина, то игра прекращается, игрок считается проигравшим, если нет — то в клетке отображается количество соседних мин. В случае, когда мин рядом нет, то вскрывается данная клетка, а также все соседние клетки. Игрок может помечать клетки, где могут находиться мины, не вскрывая их. Цель игрока — обнаружить все мины, то есть вскрыть все клетки без мин. Требуется написать программу «Сапер», играющую за игрока. Ясно, что не всегда удается определить ход, гарантированно не приводящий к проигрышу. Однако можно вычислить наиболее вероятный успешный из всех возможных ходов.

17.77. Спички — не игрушка! В игре участвуют два игрока. Первоначально в кучке 100 спичек. Игроки поочередно берут по несколько спичек: не менее одной и не более десяти. Проигрывает тот игрок, который берет последнюю спичку. Написать программу, играющую за одного из игроков. Вариант игры: до начала игры можно определить верхний предел количества спичек, которые забираются за один ход игрока.

17.78. Сама себя. Написать программу, выводящую свой собственный текст. Ограничение: тривиальные программы, использующие открытие файла с исходным текстом или некоторые особенности конкретного языка (например, в Бейсике программа `10 LIST`), решением не считаются.

17.79. Все соседи разные. Дано натуральное число $N (2 \leqslant N \leqslant 10)$. Заполнить квадратную матрицу порядка N целыми числами $0, 1, 2, 3$ таким образом, чтобы у любого элемента матрицы все его соседи были различны (соседом элемента считается другой элемент, у которого либо номер строки, либо номер столбца отличается на единицу).

17.80. Пересечение отрезков. Дано множество из n отрезков числовой прямой, задаваемых координатами концов a_i, b_i, где $a_i \leqslant b_i, i = 1, 2, \ldots, n$, a_i и b_i– действительные числа, $1 \leqslant n \leqslant 100$. Дано действительное положительное число l. Найти наименьшее число m $(0 \leqslant m < n)$, такое, что при удалении некоторых m отрезков из множества пересечение оставшихся будет иметь длину, не меньшую чем l, либо сообщить, что такого числа не существует.

Пример.

Введите n=*4*

Введите попарно концы 4 отрезков:

0 2

1 3

2 4

0 4

Введите *l*=*2*

Ответ: минимальное количество отрезков, которые надо удалить, равно 2

17.81. Объединение отрезков. Дано множество из n отрезков числовой прямой, задаваемых координатами концов a_i, b_i, где $a_i \leqslant b_i, i = 1, 2, \ldots, n$, a_i и b_i– действительные числа, $1 \leqslant n \leqslant 100$. Дано действительное положительное число l. Найти наименьшее число $m (0 \leqslant m < n)$, такое, что при удалении некоторых m отрезков из множества объединение оставшихся будет иметь длину, не большую чем l, либо сообщить, что такого числа не существует. Длиной объединения множества отрезков будем считать сумму длин частей, из которых состоит это объединение. Например, длина объединения отрезков $[0; 3]$, $[1; 2]$ и $[4; 5]$ равна 4.

Пример.

Введите n=*4*

Введите попарно концы 4 отрезков:
0 2
1 3
2 4
0 4
Введите l=*2*
Ответ: минимальное количество отрезков, которые надо удалить, равно 3

17.82. Ориентирование на местности. Соревнования по ориентированию проходят на некоторой местности. Местность имеет форму квадрата размером 2×2 километра. На местности введена система координат, начало которой расположено в левом нижнем углу квадрата, оси направлены соответственно слева направо и снизу вверх. Квадрат разбит на два равных прямоугольных участка размерами 2×1 километр, обозначенные буквами A и B; участок A расположен внизу, участок B расположен вверху квадрата.

У участника имеется карта местности, на которой помечены n контрольных пунктов, в каждом из которых он должен отметиться. Путь участника начинается в начале координат. Между двумя любыми пунктами можно выбирать путь в виде линии любого типа, выходить за пределы квадрата метности нельзя. Так как характер местности на каждом из участков различен, то скорости, с которыми участник может двигаться по ним, могут отличаться: обозначим их v_A и v_B соответственно, скорость измеряется в метрах в минуту. При движении по границе между участками скорость движения равна скорости более «быстрого» участка из двух.

Требуется найти минимальное время, за которое участник, выйдя из начала координат, сможет обойти все контрольные пункты.

Техническое требование. В первой строке входного текстового файла указано одно натуральное число– количество контрольных пунктов n ($1 \leqslant n \leqslant 16$). Во второй строке входного файла указаны два натуральных числа– скорости передвижения участника по участкам v_A и v_B соответственно ($1 \leqslant v_A, v_B \leqslant 500$). В каждой из следующих n строк расположено по два целых неотрицательных числа x_i, y_i, обозначающих координаты контрольных пунктов в метрах ($0 \leqslant x_i, y_i \leqslant 2000$). В выходной текстовый файл требуется записать одно целое число– минимальное время в минутах, которое потребуется участнику для обхода всех пунктов, начиная с начала координат. Действительное значение времени нужно округлить до целого. Допускается ошибка не более чем на одну минуту.

17.83. Общая точка треугольников. На плоскости даны координаты n различных точек $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$, где x_l, y_l– действительные числа, $1 \leqslant l \leqslant n \leqslant 100$. Некоторые точки являются вершинами m треугольников, каждый треугольник задается номерами трех его различных вершин v_i, v_j, v_k ($1 \leqslant v_i, v_j, v_k \leqslant n$). Найти координаты любой точки плоскости, принадлежащей наибольшему числу треугольников. Точка считается принадлежащей треугольнику, если она лежит внутри него или на его границе. Числа результата вывести с тремя точными знаками после запятой.

17.84. Волшебные векторы. Назовем упорядоченный по неубыванию набор из N натуральных чисел волшебным N-вектором, если сумма этих чисел равна их произведению. Задано число N. Найдите все волшебные N-векторы.

Пример.

Введите N=*5*
Все волшебные 5-векторы:
1 1 1 2 5
1 1 1 3 3
1 1 2 2 2

17.85. Разложение на различные простые. Даны натуральные числа n, m. Найти количество натуральных чисел, не превосходящих n, таких, которые равны произведению m различных простых чисел.

Пример.

Введите n=*20*
Введите m=*2*
Ответ: таких чисел 4

17.86. Минимальный квадрат. Даны координаты n точек плоскости $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$, где x_i, y_i– действительные числа, $1 \leqslant i \leqslant n \leqslant 1000$. Найти сторону минимального квадрата, которому принадлежат все данные точки. Точка считается принадлежащей квадрату, если она лежит внутри него либо на его стороне.

17.87. Гипотеза Гольдбаха. В гипотезе Гольдбаха утверждается, что любое четное число, не меньшее 4, можно представить в виде суммы двух простых чисел. Даны натуральные числа a, b $(4 \leqslant a \leqslant b \leqslant 10^6)$. Проверить гипотезу Гольдбаха для всех четных чисел из отрезка $[a; b]$.

17.88. Теорема Л.Г. Шнирельмана. Теорема Шнирельмана утверждает, что в любую замкнутую кривую на плоскости можно вписать квадрат. Если область, ограниченная кривой невыпукла, то квадрат может выходить за границы области. Таким образом, на любой замкнутой кривой можно найти четыре точки, являющиеся вершинами квадрата.
Дана последовательность координат n различных точек плоскости $x_1, y_1, x_2, y_2, \ldots, x_n, y_n$, являющихся последовательными вершинами замкнутой ломаной линии без самопересечений. Найти квадрат максимальной площади, вершины которого принадлежат данной ломаной (то есть вершины квадрата принадлежат отрезкам ломаной).

17.89. Треугольные числа. Определим последовательность треугольных чисел $t_1, t_2, \ldots, t_i, \ldots$ следующим образом:

$$t_i = \begin{cases} 1, & \text{если } i = 1; \\ t_{i-1} + i, & \text{если } i > 1. \end{cases}$$

Таким образом, начало треугольной последовательности выглядит так: $1, 3, 6, 10, 15, 21, \ldots$
Дано натуральное число $N (N \leqslant 10^9)$. Определить, является ли оно треугольным?

17.90. Удалить цифру. Дано натуральное число $N \leqslant 10^6$. Найти основание системы счисления R $(2 \leqslant R \leqslant 36)$, чтобы при удалении одной наибольшей цифры из записи числа N в системе счисления с основанием R число уменьшилось бы максимально.

17.91. Дополнить до квадрата. Дано натуральное число $N \leqslant 10^9$. Найти минимальное простое число, при прибавлении которого к N получается полный квадрат.

Тесты, решения, подсказки

1.26. Обозначим неизвестное количество лет x. Тогда количество свечей, потраченное за x лет, равно $\dfrac{x(x+1)}{2}$ (сумма x элементов арифметической прогрессии с начальным членом 1 и разностью 1). Значит, искомое x — наибольшее натуральное число, которое удовлетворяет неравенству
$\dfrac{x(x+1)}{2} \leqslant N$. Решив его, находим x: $x = [\dfrac{-1+\sqrt{1+8N}}{2}]$, где квадратные скобки означают целую часть числа (наибольшее целое, не превосходящее данного числа).

2.28. Тесты для решения биквадратного уравнения:

N	Вход			Выход
0	1	1	0	0
1	1	-2	1	-1 1
2	1	-1	0	0 -1 1
3	1	-5	4	-1 1 -2 2
4	1	3	2	Нет корней
5	1	2	3	Нет корней

2.29. Программа дожна проходить все тесты задачи 2.28 (см.выше) и, кроме этого, следующие тесты:

N	Вход			Выход
6	0	1	-4	-2 2
7	0	20	0	0
8	0	1	100	Нет корней
9	0	0	-1	Нет корней

3.30. Очевидно, что нужно выделить общие части в факториалах и каждый раз заново их не пересчитывать. Если для нахождения коэффициента бинома Ньютона прямо применять основную формулу (см. задачу 3.25), то количество умножений будет равно $2n-2$. Если же применить оптимизацию, то оно не превзойдет $n-1$. Кроме этого, на единицу можно тоже не умножать.

6.32. Всю работу по представлению матрицы в виде «виртуального» линейного массива, который можно сортировать обычным методом сортировки, можно возложить на специальную функцию, вычисляющую координаты элемента в матрице по его «номеру» в змейке. Например, для направления расположения элементов ж) рисунка 1 можно вывести следующие формулы для координат k-го элемента змейки матрицы размера $N \times N$ (считаем, что строки и столбцы матрицы имеют индексы от 0 до $N-1$):

$$i = [k/n]; j = \begin{cases} k - i * N, & \text{, если i – четное,} \\ (i+1) * N - k - 1 & \text{, если i – нечетное.} \end{cases}$$

(Здесь квадратные скобки означают целую часть числа. Считается, что элементы змейки нумеруются подряд от нуля до $N^2 - 1$.)

Для других случаев расположения элементов матрицы формулы будут несколько сложнее – их вывод сам по себе представляет интересную математическую задачу. Этот пример показывает, что предварительная математическая проработка решения задачи может упростить алгоритм (в смысле временной сложности вычисления) на несколько порядков.

7.64. Для примера приведем цепочку превращений из «мухи» в «слона»: муха - мура - тура - тара - кара - каре - кафе - кафр - каюр - каюк - крюк - урюк - урок - срок - сток - стон - слон.

7.65. Муха - хула - луна - лунь - ноль - слон. Вообще, посмотреть различные игры такого рода можно на http://www.xword.ru.

17.27. Ответ задачи весьма прост, и его теоретическое обоснование можно найти, например, в [15] либо вывести самостоятельно.

17.28. Если вы решили предыдущую задачу, то вывести закономерность для этой не составит особого труда.

17.35. Можно воспользоваться интересным свойством чисел Фибоначчи: $\forall f_i \forall f_j : f_i \in \Phi \wedge f_j \in \Phi \Rightarrow \text{НОД}(f_i, f_j) \in \Phi$, где буква Φ обозначает множество всех чисел Фибоначчи.

17.40. Очевидно, что для чисел с таким количеством цифр обычные алгоритмы перевода из системы счисления в другую на ЭВМ практически не применимы. Программировать операции с длинными числами тоже утомительно. Гораздо проще увидеть, что для систем счисления с основаниями R и Q, где $R = Q^n$, n – целое неотрицательное число, алгоритмы перевода основаны на вполне определенных соотношениях между цифрами разных систем.

17.56. Просто надо знать (или вывести) несколько формул для многоугольника на плоскости. Интересное решение предложено М.С. Густокашиным на сайте http://algolist.manual.ru.

17.82. Для решения задачи нужно проделать некоторую предварительную работу, пользуясь методами математического анализа.

17.73. Тесты для задачи.

N	Вход	Комментарий к тесту	Выход
0	x)a(;-)(x()(	Текст примера	10
1	(f(fgr)()) ((x)((fg)(x)(	Недостаток скобок невелик	81
2	)x()xxx) ((()(c)e)0123456789)	Недостаток открывающих скобок слева	8
3	*)*)(x(x)x)*)*)*) (0(1)2(3(4)5(6)7)8)9 (+(+((x(x)x)+(+(+	Недостаток скобок и слева и справа достаточно велик	22127616
4	(((x)+(y)+((z)*5)-4)) (1.1*(6.55+(1-2)))+ (sqrt(p*(p-a)*(p-b)*(p-c)))	Большое сбалансированное выражение	0
5	(((((*)+(*))+ ((((((((*)))))))) +(((*)+(*)+(*))))	Большое почти сбалансированное выражение (недостаток двух закр. скобок)	1127
6	Пятнадцать человек на сундук мертвеца, Йо-хо-хо, И бутылка рома!	Длинный текст без скобок	0

7	**))))))))))- (((((((((	Недлинный текст с большим недостат- ком открывающих и закрывающих ско- бок	394044256
8		Пустой текст	0
9	1234567890 1234567890 1234567890 123456789) (1234567890 1234567890 1234567890 123456789	Длинный текст с недостатком одной откр. и одной закр. скобки	1600

17.74. Тесты для задачи.

N	Вход	Выход
1	-1.2(34)	-1 116/495
2	123456789	123456789
3	10.333333333	10 333333333/1000000000
4	0.098(001122)	371216/3787875
5	-333333333.789(999999)	-333333333 79/100
6	999999999.99999999(9)	1000000000
7	987654321.87654321(0)	987654321 87654321/100000000
8	-000000000.0000(00000)	0

Список рекомендуемой литературы

1. Абрамов С.А., Гнездилова Г.Г., Капустина Е.Н., Селюн М.И. Задачи по программированию. М.: Наука, 1988. 224 с.
2. Абрамов С.А., Зима Е.В. Начала информатики. М.: Наука, 1989. 256 с.
3. Пильщиков В.Н. Сборник задач и упражнений по языку Паскаль: Учеб. пособие для вузов. М.: Наука, 1989. 160 с.
4. Юркин А.Г. Задачник по программированию. СПб.: Питер, 2002. 192 с.
5. Сборник задач по базовой компьютерной подготовке/В.С.Зубов, И.Н.Котарова, О.Г.Архипов и др.; Составитель И.Н.Котарова. М.: Изд-во МЭИ, 1998. 178 с.
6. Васюкова Н.Д., Тюляева В.В. Практикум по основам программирования. Язык ПАСКАЛЬ: Учеб. пособие для учащихся сред. спец. учеб. заведений. М.: Высш. шк., 1991. 160 с.
7. Непейвода Н. Н., Скопин И.Н. Основания программирования. Москва; Ижевск, 2003. 880 с.
8. Пупышев В.В. 50 задач по началам программирования. Ижевск: Изд. дом «Удмуртский университет», 1999. 60 с.
9. Фаронов В.В. Основы Турбо Паскаля. М.: Учебно-инженерный центр «МВТУ-ФЕСТО ДИДАКТИК», 1992. 304 с. (или более поздние издания)

10. Купчинаус С.Ю., Анисимов А.Е. Программирование. Введение в алгоритмизацию задач и программирование на языке Паскаль: Учеб. пособие. Ижевск: Изд-во Удм. ун-та, 1997. 80 с.
11. Мовшович С.М., Непейвода Н.Н., Сафина А.А. Алгоритмические языки в примерах и задачах: Учеб. пособие. Ижевск: Изд-во Удм. ун-та, 1981. 162 с.
12. Кушниренко А.Г., Лебедев Г.В. Программирование для математиков: Учеб. пособие для вузов. М.: Наука, 1988. 384 с.
13. Райли Д. Абстракция и структуры данных: Вводный курс/ Пер. с англ. М.: Мир, 1993. 752 с.
14. Павловская Т.А. C/C++. Программирование на языке высокого уровня. СПб.: Питер, 2001. 464 с.
15. Воробьев Н. Н. Числа Фибоначчи. М.: Наука, 1978. 144 с.
16. Нивергельт Ю., Фаррар Дж., Рейнголд Э. Машинный подход к решению математических задач. М.: Мир, 1977. 322 с.
17. Косневски Ч. Занимательная математика и персональный компьютер/ Пер. с англ. М.: Мир, 1987. 192 с.
18. Дьяконов В.П. Справочник по алгоритмам и программам на языке бейсик для персональных ЭВМ: Справочник. М.: Наука, 1989. 240 с.
19. Уилсон Р. Введение в теорию графов. М.: Мир, 1977. 208 с.
20. Касаткин В.Н., Верлань А.Ф., Переход И.А. Элементы кибернетики школьнику (Сборник упражнений и задач). Киев: Радянська школа, 1974. 112 с.

Содержание

Предисловие 3

Задачи 5

1. Выражения. Оператор присваивания 5
2. Разветвление 11
3. Цикл с заранее известным числом повторений 17
4. Цикл с условием 25
5. Линейные массивы 33
6. Матрицы 39
7. Строки 47
8. Множества 58
9. Записи (структуры) 61
10. Подпрограммы 65
11. Файлы 71
12. Рекурсия 78
13. Динамические структуры данных. Линейные списки 86
14. Бинарные деревья, графы 95
15. Обработка текстов 107
16. Игры 109
17. Задачи на разные темы 113

Тесты, решения, подсказки 141

Список рекомендуемой литературы 145

Анисимов Андрей Евгеньевич

СБОРНИК ЗАДАЧ ПО НАЧАЛАМ ПРОГРАММИРОВАНИЯ

Редакторы, корректоры: Л.М. Клименко, В.И. Бацекало, Т.И. Чукавина, М.А. Балтин, Л.Н. Плетнева

Компьютерный набор и верстка А.Е. Анисимов

Подписано в печать 10.11.04. Формат 60×84 1/16.
Уч. печ. л. 8,6. Уч.-изд. л. 7,3.
Тираж 150 экз. Заказ № 1950.

Редакционно-издательский отдел УдГУ
Типография УдГУ
426034, г. Ижевск, ул. Университетская, д.1., корп. 4.