1.9. ЗАДАЧИ ПОВЫШЕННОЙ СЛОЖНОСТИ

1. ***«Конь и ферзь».*** На шахматной доске размером $N \times N$ клеток (N не превышает 100) находятся белый шахматный конь и черный ферзь. Составить программу определения, за какое минимальное количество ходов конь может дойти до заданной клетки, при условии, что

а) коню нельзя становиться на клетки, которым угрожает вражеский ферзь;

б) конь может «сбить» ферзя и дальше ходить свободно.

2. ***«Поля и камни».*** Прямоугольное поле состоит из $N \times M$ квадратных участков (N, M не превышают 100) двух типов — «поле» и «камень». Крестьянину надо вспахать наибольшее количество участков типа «поле» таким образом, чтобы сохранился путь из левого верхнего угла поля в правый нижний. Под путём понимается последовательность участков невспаханного поля, соседние элементы которой имеют общую сторону. Составить программу определения наибольшего количества участков, которые можно вспахать.

3. ***«Путь прямоугольника».*** Прямоугольное поле состоит из $N \times M$ квадратных клеток (N и M не превышают 100), некоторые из которых являются недоступными. В левом верхнем углу поля находится прямоугольник размером $A \times B$ клеток. Составить программу нахождения наименьшего количества шагов, за которое можно этот прямоугольник переместить в правый нижний угол, или вывода сообщения о том, что такое перемещение невозможно. Шагом считается перемещение прямоугольника на одну клетку вправо, влево, вверх или вниз; поворачивать его нельзя.

4. ***«Саперы 1».*** Прямоугольное поле состоит из $N \times M$ квадратных участков (N, M не превышают 100), некоторые участки заминированы. С юга и с севера поле ограничено непроходимыми болотами. Саперам необходимо разминировать как можно меньше участков таким образом, чтобы освободить путь для войск с востока на запад (войска могут переходить с участка на участок, если последние имеют общую сторону). Составить программу определения количества участков, которые надо разминировать.

5. ***«Саперы 2».*** Прямоугольное поле состоит из $N \times M$ квадратных участков (*N, M* не превышают 100), некоторые участки заминированы. Для разминирования этого поля было направлено несколько саперов, которые должны работать независимо. Каждому из них была выделена связная заминированная часть поля (множество заминированных участков считается связным, если существует путь между двумя любыми его участками, принадлежащих этому же множеству; путь — это последовательность участков, соседние элементы которой имеют общую сторону). Составить программу:

а) определения количества саперов, которое необходимо для разминирования этого поля;

б) вычисления коэффициента несправедливости, который равен отношению наибольшей и наименьшей из площадей, выделенных для разминирования одному саперу.

6. «Лабиринт». Лабиринт состоит из $N \times M$ квадратных участков (*N*, *M* не превышают 100), некоторые его участки недоступны. Ходить можно только по свободным участкам, перемещаясь с одного на другой при наличии у них общей стороны. Известно, что существует путь через лабиринт справа налево. Составить программу определения минимального количества участков, которые надо сделать недоступными, чтобы прервать этот путь.

7. ***«Урожай».*** Прямоугольное поле состоит из $N \times M$ квадратных участков (*N*, *M* не превышают 100). В результате боевых действий всё поле было заминировано. Саперы в произвольном порядке разминировали некоторые участки, а потом прервали свою работу. Крестьянам надо при помощи комбайна собрать урожай с как можно большего количества незаминированных участков при условии, что комбайн может заезжать на поле с любой стороны и передвигаться по незаминированным участкам, перемещаясь с одного на другой при наличии у них общей стороны. Составить программу определения количества участков, с которых крестьяне могут собрать урожай.

8. *«Этаж».* На некотором этаже дома размещаются N рядов из M-квадратных комнат (N, M не превышают 100). Каждая комната имеет двери во все соседние комнаты (от 2 до 4 штук), но некоторые из них не работают. В одной из комнат находится лифт на нижние этажи. Составить программу определения количества комнат, в которые можно дойти из комнаты с лифтом.

9. *«Шоссе».* Прямоугольное поле состоит из $N \times M$ квадратных участков (N, M не превышают 100). На некоторых участках поля растут деревья. Руководство района решило построить шоссе, которое начинается на левой стороне поля, а заканчивается на правой (под шоссе понимается последовательность участков, соседние элементы которой имеют общую сторону). Составить программу определения, можно ли проложить шоссе по свободной части поля так, чтобы оно делало повороты не менее, чем через K участков. Если да, то найти кратчайшую длину такого шоссе.

10. *«Лестница».* Прямоугольное поле состоит из $N \times M$ квадратных участков (N, M не превышают 100). Над каждым участком построена колонна некоторой высоты. Путем («лестницей») порядка K будем считать последовательность участков, соседние элементы которой имеют общую сторону и колонны, различающиеся по высоте не более, чем на K.

Составить программу определения:

а) длины кратчайшего пути, порядок которого не превышает заданного *K,* между левым верхним участком поля и правым нижним;

б) кратчайшей длины пути наиболее низкого порядка между левым верхним участком поля и правым нижним.

11. ***«Сейф».*** Банковский сейф открывается с помощью двух дисковых переключателей. Дисковый переключатель – это диск с рычажком посредине, который может поворачиваться по кругу, занимая *K* (*K* не превышает 10) разных положений, отмеченных делениями. Сначала рычажки установлены в положение «вверх». За одно движение любой рычажок можно повернуть на 1 или 2 деления по или против часовой стрелки. При этом другой рычажок автоматически поворачивается на некоторое известное количество делений (на 1 или 2, по или против часовой стрелки). Сейф открывается, когда рычажки занимают определенные положения. Грабителям удалось узнать, какие это положения. Остается только успеть открыть сейф. Составить программу определения минимального количества движений, открывающих сейф.

12. ***«Квадрируемый прямоугольник».*** Составить программу определения, является ли заданный прямоугольник размером $N \times M$ (N, M — натуральные числа, не превышающие 200) квадрируемым. Прямоугольник называется квадрируемым, если его можно разрезать на несколько *разных* квадратов, длины сторон которых выражаются натуральными числами.

Например*, прямоугольник 33 × 32 является квадрируемым, его можно разрезать на 9 квадратов со сторонами: 18, 15, 14, 10, 9, 8, 7, 4, 1.*

13. ***«Телебашни».*** В некотором государстве существует столица и N городов (N натуральное число, не превышающее 50), расположение которых задано. В столице установлены телестудия и телебашня, радиус действия которой R. Руководство решило расширить телевизионную сеть с наименьшими затратами, разместив такие же телебашни еще в некоторых городах. Город считается обеспеченным телесигналом, если он находится в радиусе действия любой телебашни. На телебашню поступает сигнал, если она находится на расстоянии не больше L от телебашни, уже обеспеченной сигналом. Составить программу определения наименьшего количества телебашен, которые необходимо установить для обеспечения сигналом всех городов.

14. ***«Гаражи».*** Прямоугольное поле можно условно разбить на $N \times M$ квадратных участков со стороной 1 *м* (N и M не превышают 50). Некоторые из участков поросли деревьями. Составить программу определения наибольшего количества прямоугольных гаражей размером $P \times K$ метров, которые можно разместить на поле, не срубив ни одного дерева.

15. ***«Переправа».*** На одном берегу реки стоят M человек (M не превышает 50) и одна лодка, которая вмещает только двух человек. Для каждого человека задано время, за которое он может переправиться через реку. Если в лодке сидят два человека, то время переправы определяется как наибольшее из двух. Составить программу определения наименьшего времени, за которое все эти люди могут переправиться на другую сторону реки.

Например*, для данных 10, 5, 2, 1 общее кратчайшее время переправы будет составлять 17.*

16. ***«Игра Сапер".*** В известной игре "Сапер" на прямоугольном клеточном поле размером $N \times M$ (N и M не превышают 20) размещены несколько мин. В каждой свободной клетке поля записывается число от 0 до 8, которое равно количеству мин в соседних клетках. Составить программу отыскания расположения мин на поле по заданным количествам клеток, содержащих, соответственно, запись «0», «1», …, «8». Если это невозможно, выдать соответствующее сообщение.

17. «*Игра 15*». Принцип известной игры «15» таков: на клеточном поле размерами 4 × 4 в произвольном порядке находятся 15 фишек, которые занумерованы числами от 1 до 15; одна клетка остается свободной. За один ход можно переместить на свободную клетку одну из соседних фишек. Составить программу определения (если это возможно) наименьшего количества ходов, за которое можно расставить фишки по порядку (в верхней строке фишки с номерами 1, 2, 3, 4, в следующей — 5, 6, 7, 8 и так далее).

18. «*Ладья*». На шахматной доске размером *M* (*M* не превышает 20) стоит белая ладья и несколько черных фигур. Составить программу нахождения минимального количества ходов, за которые ладья может сбить все фигуры.

19. «*Пирамидки*». Из *M* дисков разного или одинакового диаметра (*M* не превышает 20) можно составлять пирамидки по такому правилу: на больший диск можно сверху класть меньший. Разработать программу определения количества разных наборов пирамидок, которые можно составить из заданных дисков, используя для каждого набора все диски.

Например*, из 3-х разных дисков можно составить 5 различных наборов.*

20. «*Соседи*». В *M*-этажном отеле (*M* не превышает 20), на каждом этаже находится по 4 одноместных номера. Отель полностью заселен постояльцами, из которых *A* «мамайцев», *B* «дубинцев», остальные — «криканцы». Представители разных групп любят друг друга, но не любят своих. Можно вычислить «коэффициент нелюбви» для каждого постояльца — суммарное количество «своих» в номерах над, под и на одном этаже с номером постояльца (это может быть целое число от 0 до 5). Составить программу определения способа расселения постояльцев, при котором суммарный «коэффициент нелюбви» является наименьшим.

21. «*Сонет*». В поселке живут *А* «одинцов», *Б* «двойцов», *В* «тройцов» и *Г* «четверцов» (общее количество жителей не превышает 50). Каждому «одинцу» для полного счастья необходимо определенное количество яблок, фейхоа и персиков. После получения необходимого количества фруктов «одинец» достигает наивысшего блаженства и посвящает своим соседям сонет. Больше одного сонета «одинец» придумать не может. Подобным образом действуют и представители групп *Б*, *В* и *Г*, только им для полного счастья необходимы другие количества тех же фруктов, и они способны от полноты чувств написать, соответственно, два, три, четыре сонета. Составить программу определения наибольшего количества сонетов, которые способные написать жители поселка, если они имеют *Я* яблок, *Ф* фейхоа и *П* персиков.

22. «*Снайпер*». На шахматной доске размерами 8 × 8 стоит специальная фигура — снайпер и несколько вражеских фигур. Снайпер может передвигаться по шахматной доске за один ход на одну клеточку в 8-ми направлениях. Кроме того, снайпер может сбить любую фигуру по вертикали, горизонтали или диагонали, не тратя на это ход. Составить программу определения наименьшего количества ходов, за которые снайпер может сбить все вражеские фигуры.

23. «*Болото*». На болоте находятся *M* круглых кочек (*M* не превышает 20) разных радиусов. На одной кочке стоит Ваня, на другой находятся ягоды. Ваня хочет, прыгая с кочки на кочку, добраться до ягод, съесть их, и возвратиться назад. Ваня может прыгать на расстояние, которое не превышает *R*. Кочка, на которой Ваня уже побывал, тонет (кроме первой). Составить программу определения наименьшего количества прыжков, которые Ваня может сделать для выполнения этой задачи.

24. «*Сокобан*». В древней Японии очень популярной была игра «Сокобан». Изложим суть игры в ее упрощенном варианте.

Квадратная игровая доска состоит из $N \times N$ клеток (N не превышает 50), некоторые из которых недоступны. Игроку предоставляется две фишки: белая и красная. Белая фишка стоит на заданной клетке, начальное положение красной игрок избирает самостоятельно. Задача игрока состоит в том, чтобы привести белую фишку в клетку (1;1) за наименьшее количество шагов (одним шагом считается переход из клетки на одну из четырех ее соседних клеток: вперед, назад, влево или вправо). Сделать шаг белая фишка может лишь в том случае, если ее с противоположной стороны подталкивает красная фишка (при этом красная фишка тоже перемещается на одну клетку). Красную фишку игрок может переставлять по доске также только по доступным клеткам.

Составить программу определения решения задачи игрока для заданной доски и начального положения белой фишки:

а) учитывая только ходы белой фишки;

б) учитывая только ходы красной фишки;

в) учитывая ходы белой и красной фишек.

25. «*Золотые кубики*». Прямоугольный параллелепипед размерами $N \times M \times K$ (N, M, K не превышают 10) состоит из одинаковых по форме железных и золотых кубиков со стороной 1. Составить программу определения наибольшего количества золотых кубиков, которые можно вытянуть из стоящего на полу параллелепипеда так, чтобы при этом другие кубики не упали. Кубик не падает, если он стоит на полу или соединен с кубиком, стоящим на полу, через цепочку других кубиков, имеющих попарно общие грани.

26. «*Клеточные многоугольники.* Разработать программу для нахождения оптимальной склейки двух заданных клеточных многоугольников.

Клеточным многоугольником называется плоская фигура, ограниченная замкнутой ломанной без самопересечений, звенья которой параллельны осям декартовой прямоугольной системы координат, а вершины имеют целые координаты.

Склейкой клеточных многоугольников называется клеточный многоугольник, который можно образовать из этих многоугольников путем их параллельного перенесения и поворота и без наложения друг на друга.

Склейка является оптимальной, если имеет минимальное возможное количество сторон.

27. «*Склад*». Воинский склад разбит на $M \times N$ квадратных участков (M, N не превышают 20). На складе имеется P контейнеров, каждый из которых занимает 1 участок. Контейнеры вывозятся со склада через эвакуатор, расположенный на угловом участке склада (1;1). При перемещении контейнера к эвакуатору он перетягивается с одного участка на другой, свободный, имеющий с ним общую сторону. Если контейнер оказывается заставленным другими контейнерами, то солдатам надо расчистить для него путь, отставив эти контейнеры в сторону или даже убрав их со склада через эвакуатор. Однако потом их нужно снова поставить на место.

Для каждого контейнера может быть рассчитана его трудоемкость как сумма всех элементарных перемещений, которые нужно выполнить для его эвакуации. Составить программу отыскания такой расстановки контейнеров на складе, при которой суммарная трудоемкость всех контейнеров минимальна.

28. *«Ограда».* Прямоугольное поле состоит из $M \times N$ квадратных участков (M, N не превышают 50) со стороной P метров. Некоторые участки поля отданы под дачи, остальные свободны. Дачный кооператив решил огородить поле, однако материалов для ограды хватает только на P метров. Ограду можно проводить только по границам участков. Составить программу построения такой ограды, которая защищает наибольшее количество занятых участков.

29. «*Шахматная доска*». Существует старинная индийская легенда: у одного раджи была золотая шахматная доска, в клетки *1a*, *2b*, *3c* и *4d* которой были вставлены огромные бриллианты. После смерти раджи четыре его сына решили разделить эту шахматную доску на четыре равных части таким образом, чтобы каждому достался один бриллиант. Только один математик смог решить эту трудную задачу, за что был удостоен большого почета.

Составить программу разделения шахматной доски (если это возможно) на четыре равных части для заданного положения бриллиантов.

30. «*Коробок*». Коробок имеет основание $M \times K$ и высоту H (M, K, H — натуральные числа, не превышающие 5). За один ход коробок можно перекатить через ребро с одной грани на соседнюю. Составить программу определения наименьшего количества ходов, за которое коробок можно перекатить в заданную позицию. В конечной позиции положение коробка является таким же, как и в начальной.

31. «*Пути*». Прямоугольное поле состоит из $N \times M$ квадратных участков, некоторые из которых являются недоступными (N, M не превышают 50). Составить программу определения количества различных путей, которые соединяют правую и левую стороны поля (путем называется последовательность квадратных участков, соседние элементы которой имеют общую сторону; пути считаются разными, если они не имеют общих участков).

32. «*Надежное электроснабжение*». *N* городов объединено электросетью (*N* не превышает 50). Звенья сети объединяют каждый город с остальными. В первом городе находится источник электроэнергии. Электроснабжение считается надежным степени *K* (*K* меньше *N*), если каждый город подсоединен к источнику по меньшей мере К независимыми линиями (не имеющими общих звеньев). Составить программу определения наименьшего количества звеньев, которые надо оставить, чтобы обеспечить заданную степень надежности.

33. «*Железная дорога*». На железной дороге, которая соединяет два города, есть *M* станций, расстояния между которыми заданы. Между этими станциями курсируют *M* поездов (все поезда направляются от начальной станции к конечной без остановок). С целью сокращения затрат руководство железной дороги решило уменьшить количество поездов до минимума, с тем лишь условием, чтобы возможно было добраться из каждой станции к каждой (возможно, с пересадками). Составить программу определения такой совокупности поездов, для которых суммарная длина маршрутов является наименьшей.

34. «*Отрезки*». *N* отрезков на прямой заданы целочисленными координатами своих концов (*N* не превышает 100). Составить программу определения среди этих отрезков *K* таких, которые покрывают по возможности больший участок прямой.

35. «*Кинотеатр*». В некотором городе функционируют *N* кинотеатров. В каждом кинотеатре демонстрируется один фильм, время начала и окончания показа которого заданы (начало — после 0^{00}, конец — до 24^{00}). Составить программу определения, какое наибольшее число фильмов можно посмотреть целиком на протяжении суток (начиная с 0^{00}), если время, необходимое для того, чтобы добраться от одного кинотеатра к другому, является постоянным и равно K.

36. «*Путь круга*». Центр круглого тела радиуса R находится в точке $(X;Y)$. Перемещая тело, нужно совместить его центр с началом координат. Сложность состоит в том, что на плоскости имеются K запрещенных точек (K не превышает 50), которые тело заслонять собой при перемещении не может (разрешено только касание). Составить программу определения длины кривой, которую описывает центр круга при перемещении тела по кратчайшей траектории.

37. «*Цепочка*». На плоскости задано N точек своими координатами (N не превышает 20). Надо соединить точки ломаной, у которой все углы прямые и все звенья параллельны осям координат. Составить программу определения ломаной с наименьшим количеством вершин.

38. «*Оплётка*». Составить программу определения, оплетает ли заданная пространственная ломаная некоторый прямоугольный параллелепипед. Считается, что ломаная оплетает фигуру, если все ее звенья принадлежат граням этой фигуры.

39. «*Объединение*». Множество N ($N \leq 1000$) отрезков $[a_i; b_i]$ на прямой задано координатами своих концов. Составить программу определения объединения этих отрезков.

40. «*Ломаная*». Ломаная на плоскости задана координатами своих N вершин (N не превышает 100). Составить программу определения:

а) имеет ли ломаная самопересечения;

б) существует ли луч AB с началом в точке $A(x, y)$, который не пересекает ломаную;

в) существует ли отрезок AB длиной h с началом в точке $A(x, y)$, который не пересекает ломаную;

г) расстояния от точки $A(x, y)$ до ломаной;

д) наименьшего количества звеньев ломаной, которые необходимо удалить для того, чтобы она не имела самопересечений;

е) минимального количества отрезков, удаление которых приводит к нарушению связности ломаной;

ж) зигзагообразна ли ломаная (не имеет самопересечений, повороты по и против часовой стрелки чередуются);

з) наименьшего количества соединений вершин ломаной отрезками, необходимого для получения фигуры, состоящей лишь из треугольников;

и) можно ли получить выпуклый многоугольник путем удаления нескольких звеньев ломаной.

41. «*Динг-донг*». В некоторой южной стране очень популярной является игра в динг-донг. Для этой игры необходимо чистое поле в виде выпуклого N-угольника, в вершинах которого расположены пальмы. Составить программу определения поля наибольшей площади для заданного размещения пальм.

42. «*Дорога*». На прямоугольном участке размером $X \times Y$ метров посажено H деревьев радиуса R. Составить программу определения дороги наибольшей ширины, которая бы соединяла левую и правую стороны участка, минуя деревья. Дорогой считается множество точек, которые содержатся между двумя параллельными отрезками, одни концы которых принадлежат левой, другие — правой сторонам участка; ширина дороги — это расстояние между отрезками.

43. «*Изгородь 2*». Прямоугольное поле состоит из $N \times M$ квадратных участков (N, M не превышают 100). На некоторых участках построены здания. Составить программу нахождения ограды наименьшей длины, которой можно обнести все здания. Изгородью считается замкнутая последовательность участков, которые попарно касаются сторонами или вершинами (длина изгороди — это количество участков).

44. «*Учитель*». Некоторый учитель математики придумал игру «Угадай число». Правила этой игры такие: учитель задумывает произвольное натуральное число (которое не превышает 1 млрд.), переводит его в 2, 3, 4, ..., P-ичные системы счисления (P<10), находит суммы цифр полученных чисел (в десятичной системе счисления) и сообщает их ученикам, которые должны по этим суммам отгадать задуманное число. Составить программу поиска всех чисел, которые соответствуют заданным суммам.

45. «*Геометрическая прогрессия*». Составить программу определения, можно ли из элементов заданной линейной таблицы образовать геометрическую прогрессию путем их некоторой перестановки (геометрическая прогрессия — это последовательность чисел a_1, a_2, …, a_{n}, где $a_{n+1} = d \cdot a_n$ ($d \neq 0$, $|d| \neq 1$)).

46. «*Арифметическая прогрессия*». Задана линейная целочисленная таблица, упорядоченная строго по возрастанию. Составить программу определения минимального количества элементов, которые необходимо добавить в эту таблицу, чтобы образовалась арифметическая прогрессия (арифметическая прогрессия — это последовательность чисел a_1, a_2, …, a_n, где $a_{n+1} = d + a_n$ ($d \neq 0$)).

47. «*Путь*». В прямоугольной таблице размерами $M \times N$ (M, N не превышают 100) записаны цифры. Составить программу определения такого пути из ячейки (1;1) (левый верхний угол таблицы) в ячейку (M;N) (правый нижний угол таблицы), чтобы сумма элементов таблицы, которые стоят на этом пути, была минимальной. Ходить разрешается только вниз и вправо.

48. *«Многоугольник».* M точек на плоскости заданы своими координатами (M не превышает 100). Составить программу определения последовательности соединения этих точек для получения многоугольника.