

Задания для 5–8 класса

Заключительный этап (приведен один из вариантов заданий)

1. Кодирование информации, информационный объем (1 балл)

[Старый плеер]

Даша нашла на старом плеере три музыкальные композиции и заметила интересную особенность: все композиции имеют одинаковую частоту дискретизации, однако глубина кодирования первой композиции в 2 раза меньше глубины кодирования второй композиции, а глубина кодирования третьей – в 3 раза больше глубины кодирования второй. Тогда Даша придумала задачку и предложила ее решить своей подруге Алисе: определить длительность третьей композиции. Все композиции двухканальные, а также известно, что первая и третья композиции имеют одинаковый информационный объем, а длительность первой композиции равна 120 секундам. Помогите Алисе решить задачку Даши. В ответ запишите одно целое число – длительность третьей композиции в секундах.

2. Комбинаторика (2 балла)

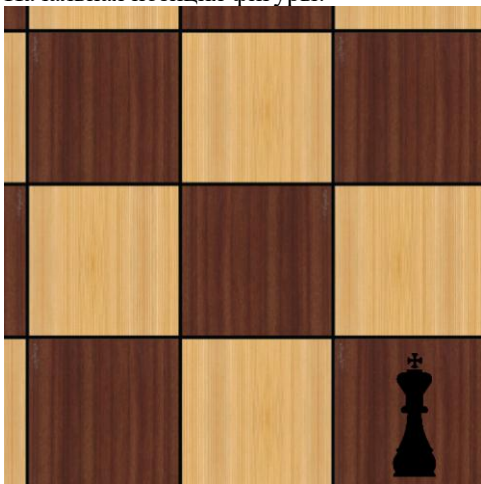
[Шахматы 2: Спасти рядового Пешку]

Дана шахматная доска размера 16x16 клеток и фигура, которая может ходить только по диагонали, но со следующими ограничениями:

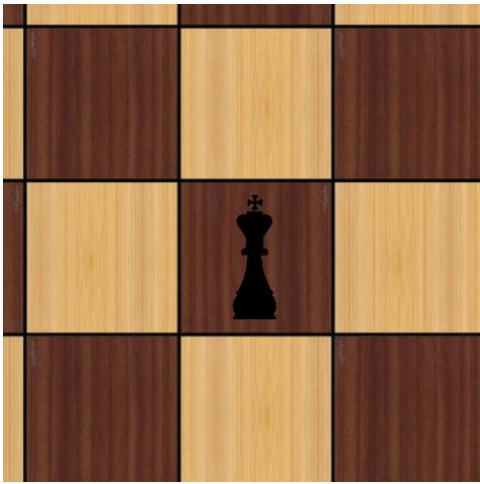
- 1) Фигура не может ходить по диагонали вправо-вниз.
- 2) На ходах с нечётным номером фигура может пойти **только** влево-вверх.
- 3) На ходах с чётным номером фигура может выбрать **одно из двух** направлений:
 - a. Пойти вправо-вверх.
 - b. Пойти влево-вниз.
- 4) Нумерация ходов начинается с единицы.

Пример:

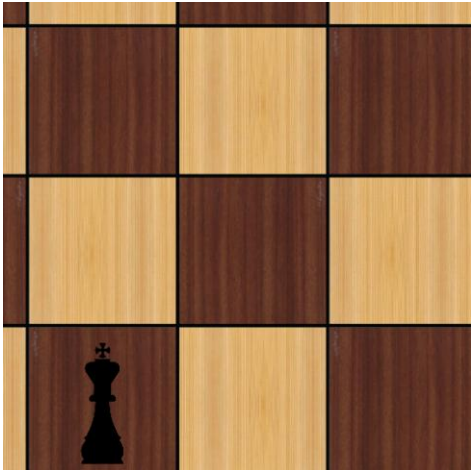
Начальная позиция фигуры:



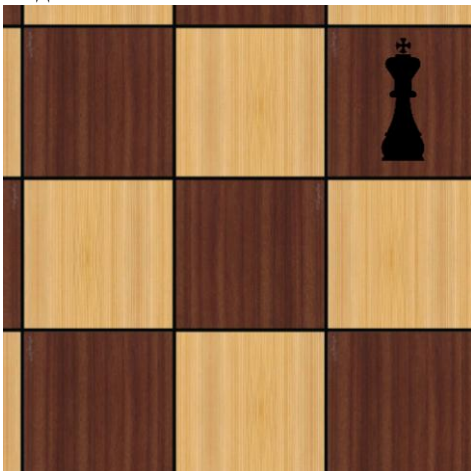
Ход №1:



Ход №2а:

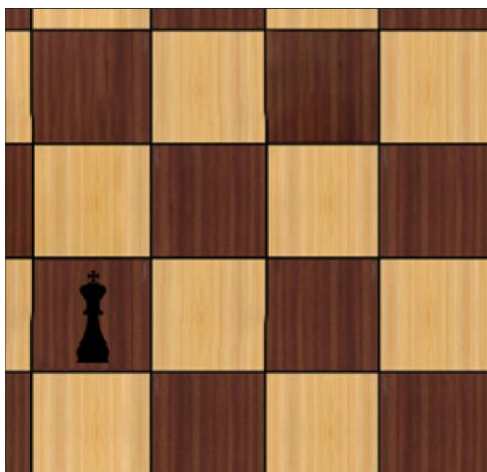


Ход №2б:

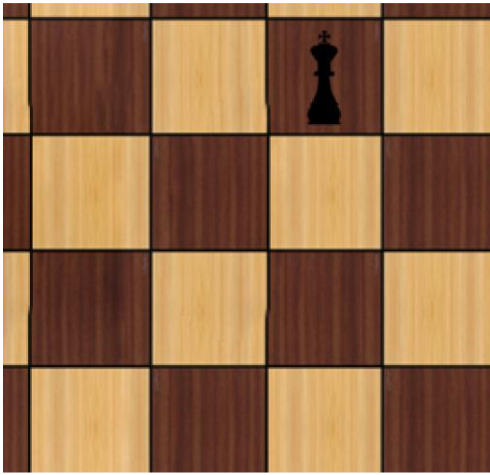


Ходы №2а и №2б являются взаимоисключающими.

Ход №3а (из позиции хода №2а):



Ход №3б (из позиции хода №2б):



Сколько существует различных траекторий, отличающихся хотя бы одним ходом, для того, чтобы добраться из правого-нижнего угла доски в левый-верхний угол? Число возможных траекторий напишите в ответ.

3. Системы счисления. Теория игр (2 балла)

[Игра в числа]

Прохор и Василиса придумали игру с числами.

В начале игры на доске записывается некоторое стартовое число в десятичной системе счисления. Далее игроки ходят по очереди. Каждый игрок в свой ход стирает число на доске и записывает вместо него запись этого числа в восьмеричной или в девятеричной системе счисления (на свой выбор). Запись на доске следующим игроком читается как новое десятичное число. Выигрывает игрок, записавший в свой ход на доске «300» или большее число. Первым ходит Прохор. Какое минимальное стартовое число должно было быть на доске, чтобы гарантированно выиграл Прохор своим вторым ходом? В ответ запишите искомое число в десятичной системе счисления.

Пример игры, если бы стартовым числом было 100, а для победы было необходимо записать число, равное или большее 200:

Прохор своим первым ходом переводит 100 в восьмеричную систему счисления и заменяет число на доске на 144. Василиса своим первым ходом переводит 144 в девятеричную систему счисления и заменяет число на доске на 170. Прохор своим вторым ходом переводит 170 в девятеричную систему счисления, заменяет число на доске на 208 и побеждает.

4. Основы логики (3 балла)

[Булева функция]

Петя придумал следующую булеву функцию:

$$(p \rightarrow (\text{НЕ}(q) \text{ ИЛИ } r)) \text{ ИЛИ } \text{НЕ}(r) \text{ ИЛИ } (p \text{ И } \text{НЕ}(q \rightarrow r)) \text{ ИЛИ } (q \rightarrow (\text{НЕ}(p) \text{ ИЛИ } r))$$

Также Петя знает о существовании функций, обладающих следующим свойством:

$$f(\text{НЕ}(x_1), \dots, \text{НЕ}(x_n)) = \text{НЕ}(f(x_1, \dots, x_n))$$

Теперь Петя хочет узнать, сколько у него есть способов сделать из своей функции новую, обладающую этим свойством. Новую функцию Петя может сделать следующим образом: записать новую переменную, затем логическую операцию, затем придуманную им исходно функцию в скобках. Причем новая переменная должна быть из следующего множества: {p, q, r, НЕ(p), НЕ(q), НЕ(r)}, а логическая операция из множества: {И, ИЛИ, →}.

*Пример: если изначальная функция была **p ИЛИ q**, то одним из примеров новой функции будет: **p И (p ИЛИ q)**.*

В ответе нужно записать целое число – количество способов составления новой функции, обладающей указанным свойством.

Примечание: → - ИМПЛИКАЦИЯ, логическая операция, таблица истинности которой выглядит следующим образом:

p	q	p→q
0	0	1
0	1	1
1	0	0
1	1	1

5. Кодирование информации, структуры данных (3 балла)

[Шифрование на графе]

Бобу нужно было передать пароль двум друзьям. Известно, что пароль состоит только из букв русского алфавита, причем в пароле нет повторяющихся букв.

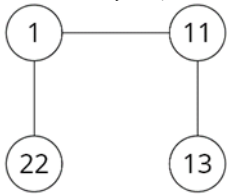
Боб не хочет, чтобы его пароль узнал кто-то посторонний, поэтому придумал для каждого из друзей различные шифры – «шифр в глубину» и «шифр в ширину».

Но для построения шифра нужно сначала построить граф пароля. Строится он следующим образом:

1. Каждая буква в пароле заменяется на число от 1 до 33 в соответствии с ее порядковым номером в русском алфавите.

2. Строится полный граф (т.е. граф, в котором каждая вершина соединена с каждой), в котором в качестве вершин выступают полученные числовые значения букв пароля.
3. Из полученного графа удаляются ребра, которые соединяют между собой те числа, буквенные значения которых НЕ стоят рядом в пароле.

Пример: «ФАЙЛ» - исходный пароль => «22 1 11 13» - численное представление. Тогда граф исходного пароля будет выглядеть следующим образом:



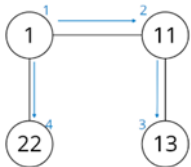
Теперь по этому графу можно построить два шифра. Для построения обоих шифров в качестве начальной вершины выбирается вершина с наименьшим номером. Соседями текущей вершины называются такие вершины, которые соединены с текущей вершиной ребром.

Для построения «шифра в глубину» необходимо:

0. Изначально все вершины графа считаются не посещенными.
1. В качестве первой текущей вершины выбрать вершину с наименьшим числом и записать ее. Считаем эту вершину посещенной.
2. Далее из соседей текущей вершины выбирается вершина с наименьшим номером и записывается ее номер. Переходим в эту вершину (т.е. она становится текущей) и считаем эту вершину посещенной.
3. Повторяем действие 2, пока не встретим вершину, у которой нет не посещенных соседей.
4. Переходим к первой посещенной вершине и повторяем действие 2 для не посещенных соседей.
5. Далее переходим ко второй посещенной вершине и повторяем действие 2 для не посещенных соседей.
6. И так далее идем по посещенным вершинам в порядке их посещения и повторяем действие 2 для не посещенных соседей, пока в графе все вершины не станут посещенными.

Для пароля «ФАЙЛ» построение и сам «шифр в глубину» будет выглядеть следующим образом:

1 11 13 22

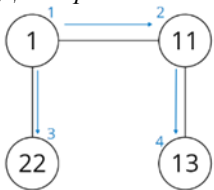


Для построения «шифра в ширину» необходимо:

0. Изначально все вершины графа считаются не посещенными.
1. В качестве первой текущей вершины выбрать вершину с наименьшим числом и записать ее. Считаем эту вершину посещенной.
2. Далее записываем всех ее не посещенных соседей в порядке возрастания чисел, записанных в этих вершинах. В этом же порядке считаем эти вершины посещенными.
3. Далее идем по посещенным вершинам (в порядке их посещения) и повторяем действие 2.
4. Идем так, пока в графе все вершины не станут посещенными

Для пароля «ФАЙЛ» построение и сам «шифр в ширину» будет выглядеть следующим образом:

1 11 22 13



Ева смогла получить оба шифра Боба и узнать какой шифр соответствует какому алгоритму:

Шифр в глубину: 10 14 16 25 20 33

Шифр в ширину: 10 14 20 16 33 25

Также Ева узнала, что пароль начинается с буквы Я.

Помогите Еве разгадать пароль Боба.

В ответ запишите пароль заглавными буквами без пробелов.

Примечание: русский алфавит:

А	Б	В	Г	Д	Е	Ё	Ж	З	И	Й	К	Л	М	Н	О	П	Р
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18
С	Т	У	Ф	Х	Ц	Ч	Ш	Щ	Ъ	Ы	Ь	Э	Ю	Я			

6. Теория игр (2 балла)

[Весенняя уборка]

Однажды, Муми-тролль вместе со Сниффом решили убраться в доме после долгой зимней спячки. К своему удивлению, они нашли так много грецких орехов, что сложили из них целую кучку. С этой кучкой они решили сыграть в следующую игру:

Игра состоит из раундов. В каждом раунде сначала ходит первый игрок, а потом второй. Каждый из игроков в свой ход может взять из кучи либо 7 орехов, либо t орехов (или все оставшиеся, если орехов в куче меньше 7 или меньше t). Игрок, после хода которого, в кучке не осталось орехов считается победителем.

В первом раунде $t = 2$. Но с каждым следующим раундом $t = t \cdot 2$. То есть во втором раунде $t = 4$, в третьем $t = 8$ и так далее.

Определите, какое минимальное число орехов изначально могло быть в кучке, чтобы гарантированно победил 1-ый игрок и при том на четвёртом раунде.

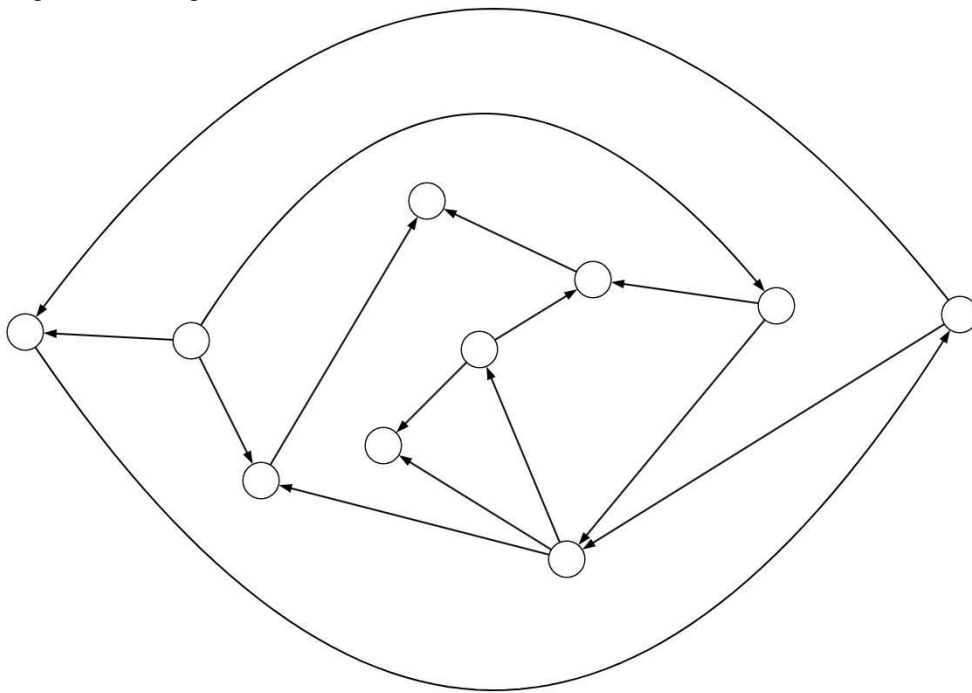
7. Моделирование на графе (1 балл)

[Музейная инновация]

Архитекторы решили оснастить музейные залы самодвижущимися дорогами - траволаторами. По такой дороге возможно двигаться лишь в одном направлении. К сожалению, из-за просчётов в плане, некоторые из дорог развернули не в ту сторону. Вы можете поменять направление каких-то дорог так, чтобы независимо от того, из какой комнаты посетитель попадает на траволатор, он мог добраться по траволатору до любой комнаты.

План музея представляет собой ориентированный граф, где комнаты обозначаются вершинами, а дороги траволатора – рёбрами. При перемещении по музею, по траволатору можно двигаться только в направлении указанному стрелкой.

Требуется исправить план музея так, чтобы число перенаправленных дорог было минимально. В ответе укажите число перенаправленных дорог.



8. Моделирование (1 балл)

[Алгоритм на таблице]

Расстоянием между числами является модуль их разности, т.е. разность между большим и меньшим числом.

Ближайшим числом к текущему числу называется такое число, что расстояние между текущим и этим числом минимально.

Дана матрица размера 10×10 , заполненная числами от 1 до 2^8 :

40	151	91	201	127	188	167	26	47	60
86	201	173	61	33	67	3	178	57	57
116	139	187	174	252	209	184	27	84	152
85	250	103	238	97	103	84	188	50	251
158	243	33	182	90	115	46	164	111	100
45	88	48	244	130	25	220	135	195	190
12	114	69	76	176	52	206	220	117	49
157	91	74	104	154	243	156	113	37	66
166	74	210	188	43	115	41	29	171	129

224	221	165	60	95	200	39	39	21	181
-----	-----	-----	----	----	-----	----	----	----	-----

К ней применяется следующий алгоритм:

1. Из отрезка $[1, 2^8]$ выбираются 2^{8-i} чисел, таких что в отсортированном виде все соседние пары чисел имеют одинаковое расстояние. Числа 1 и 2^8 входят в этот список. Выбранные числа не обязательно должны быть целыми.
2. Каждое число в матрице заменяется на ближайшее число из выбранных на первом шаге чисел. Если у числа в матрице два ближайших числа из выбранных на первом шаге, то оно заменяется на меньшее из ближайших.

Алгоритм последовательно выполняется для значений i от 0 до 7 включительно.

Требуется узнать, сколько чисел 1 станет в матрице после выполнения алгоритма. В ответ запишите целое число.

9. Алгоритмизация. Анализ блок-схемы (3 балла)

[Преобразование массива]

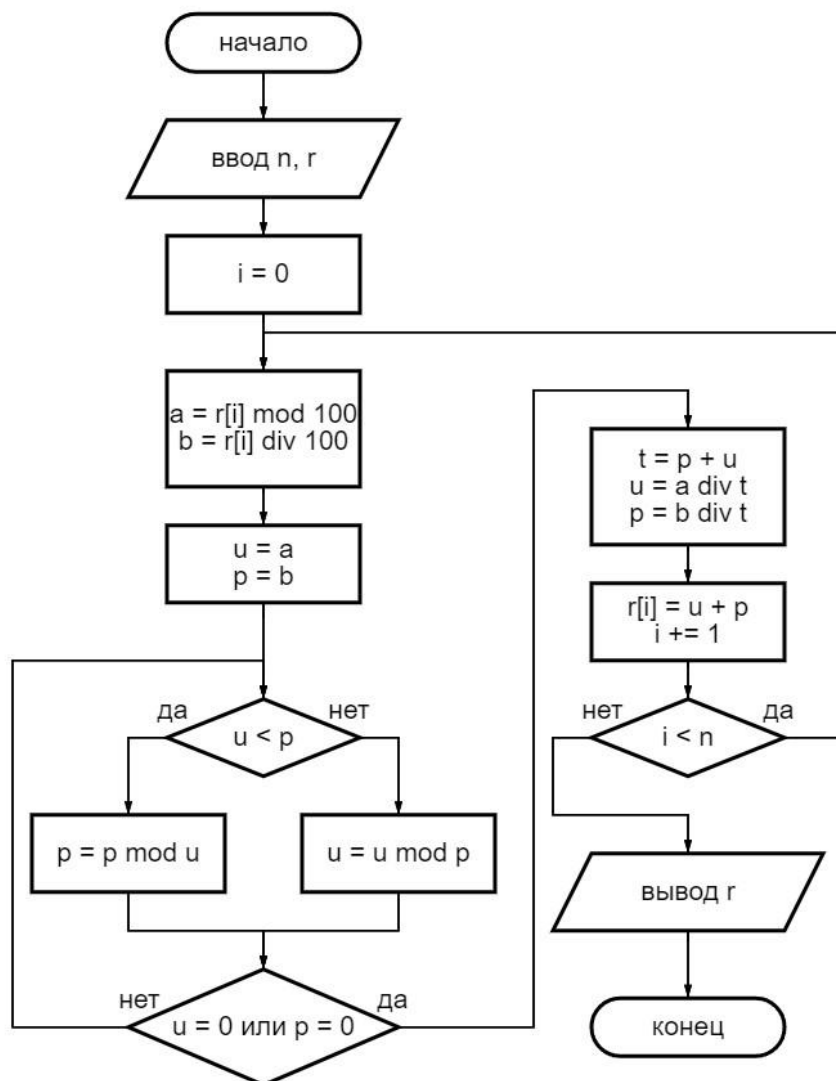
Вам дана блок-схема алгоритма, который получает на вход: число n – число элементов в массиве r и сам массив r . После алгоритм делает преобразования над элементами массива и возвращает изменённый массив r . Элементы в массиве нумеруются с нуля.

Алгоритм запустили для следующего массива ($n = 5$):

$r = \{7042, 2412, 981, 19590, 199\}$

В ответ запишите значения элементов массива после преобразований алгоритма подряд без пробелов в порядке их расположения в массиве.

(Допустим, что массив из двух элементов после работы алгоритма выглядит как: $\{44, 6\}$, тогда в ответ следует записать: 446)



Примечание:

1. " $a += b$ " – переменной a присваивается значение суммы переменных a и b .
2. " $p \bmod q$ " – взятие числа p по модулю q (остаток от целочисленного деления p на q).
3. " $p \operatorname{div} q$ " – целочисленное деление числа p на q .
4. " $r[i]$ " – обращение к элементу массива под номером i .

10. Сортировка и фильтрация данных (2 балла)

[Сортировка Шелла]

Оля реализовала новую для себя сортировку. На вход программа получает массив из N элементов и набор значений d , расположенных от больших к меньшим. На каждом шаге алгоритма между собой сравниваются и сортируются все элементы, находящиеся друг от друга на расстоянии d , после чего берётся следующее значение для d и так продолжается до тех пор, пока d не будет равен 1 (см. пример). После этого массив будет гарантированно отсортирован.

Примечание:

Элемент $A[j]$ находится на расстоянии d от элемента $A[i]$, если $A[i + d] = A[j]$ или $A[i - d] = A[j]$.

D – массив из элементов d_i . Алгоритм выполняется для всех d_i из массива D последовательно.

Пример работы сортировки: для массива $A = [7, 2, 4, 6]$ и значений $D = [3, 2, 1]$

$d = 3$: $[7, 6]; [2]; [4] \Rightarrow [6, 7]; [2]; [4] \Rightarrow [6, 2, 4, 7]$

$d = 2$: $[6, 4]; [2; 7] \Rightarrow [4; 6]; [2; 7] \Rightarrow [4, 2, 6, 7]$

$d = 1$: $[4, 2, 6, 7] \Rightarrow [2, 4, 6, 7]$

Получили $[2, 4, 6, 7]$

На вход программе подаётся:

Массив $A = [72, 26, 114, 15, 21, 39, 6, 40, 115, 13, 7]$

И значения $D = [7, 3, 1]$

Представим, что из-за сбоя алгоритм не закончил сортировку и остановился сразу перед значением $d = 1$ (выполнив сортировку для $d = 7$ и $d = 3$, но не выполнив для $d = 1$).

Какой элемент будет находиться в массиве по индексу “6” после этих преобразований? Индексы в массиве пронумерованы с нуля. В ответ запишите число.