

I. Задания заключительного этапа олимпиады 2021-22 года

Заключительный этап для 11 класса (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (3 балла)

[Двоичный ряд]

Петя строит ряд двоичных чисел следующего вида:

$0.(01)_2, 0.0(01)_2, 0.00(01)_2, \dots$

Каждый член ряда представляет собой бесконечную периодическую дробь. Для получения каждого следующего числа Петя берет предыдущее число и вставляет один ноль после разделителя целой и дробной части.

Петя взял первые 30 чисел этого ряда и сложил их, записав результат в шестнадцатеричной системе счисления. Посчитайте сумму цифр в дробной части получившегося числа. В ответе укажите целое число в десятичной системе счисления. Например, если запись суммы чисел в шестнадцатеричной системе счисления будет равна $1,ABC_{16}$, то в ответ нужно записать 33.

2. Кодирование информации. Объем информации (2 балла)

[RLE]

Петя сохраняет в память текст. Известно, что алфавит текста составляет 32 символа. Петя использует равномерное кодирование и сохраняет в память коды символов, используя минимально возможное одинаковое для кодов всех символов количество бит.

Вася и Таня изучают кодирование длин серий run-length encoding (RLE) и пытаются его применить к тексту Пети для того, чтобы он занимал меньше места в памяти.

В случае RLE кодирования весь текст представляется как расположенные друг за другом непересекающиеся последовательности из одинаковых символов так, что символы в двух соседних последовательностях отличаются. При этом в памяти сохраняется для каждой последовательности два значения: длина последовательности и код повторяющегося в ней символа. Для записи кода символа используется минимальное возможное одинаковое для кодов всех символов количество бит. Для записи длины последовательности используется минимально возможное одинаковое для всех возможных значений этого параметра количество бит.

Вася обнаружил, что максимальная длина последовательности из одинаковых символов, которая встречается в тексте, равна M и решил, что в строке могут встречаться все возможные длины последовательностей, не превышающие M . Исходя из этого предположения, он определил количество бит, необходимое для записи длины последовательности и, сохранив в память текст Пети с помощью кодирования RLE, обнаружил, что он занимает ровно 208 байт памяти.

Также Вася обнаружил, что максимальная длина последовательности из одинаковых символов ровно в X раз меньше общего количества символов в тексте Пети.

Таня проанализировала текст более внимательно и обнаружила, что количество различных длин последовательностей, которые встречаются в тексте ровно в 16 раз меньше, чем длина максимальной последовательности. Исходя из этого, она определила другое количество бит, необходимое для записи длины последовательности и, сохранив в память текст Пети с помощью кодирования RLE, обнаружила, что он занимает ровно 144 байта памяти.

Определите значение X , если известно, что текст, сохраненный Петей, занимает на 12272 байта больше памяти, чем текст, сохраненный Васей. Если таких значений несколько, определите минимальное из них. В ответе укажите целое число.

3. Основы логики (2 балла)

[Много чисел]

Логический преобразователь работает следующим образом:

1. На вход преобразователю подаётся целое неотрицательное число N , меньше 2^{64}_{10} , записанное в шестнадцатеричной системе счисления. Запись числа переводится в двоичную систему счисления и разбивается на 16 групп по 4 разряда (при необходимости в начале записи дописываются незначащие нули).
2. Двигаясь справа налево по очереди берется каждая группа двоичных разрядов и представляется как последовательность из четырех логических переменных $X_0X_1X_2X_3$, так, что единичное значение двоичного разряда соответствует значению "истина", а нулевое - значению "ложь". Например, в числе $0\dots010111$ первая такая последовательность будет $X_0 = \text{«ложь»}$, $X_1 = \text{«истина»}$, $X_2 = \text{«истина»}$, $X_3 = \text{«истина»}$.
3. Полученные последовательности логических переменных по очереди подставляются в логическое выражение:
$$(X_0 \vee X_1 \wedge X_2) \wedge (X_1 \vee X_2 \wedge X_3) \wedge (X_2 \vee X_3 \wedge X_0)$$
и вычисляется его значение.
4. Вычисленные значения преобразуются в двоичные разряды (также «истина» = 1, а «ложь» = 0) и образуют двоичную последовательность, заполняемую от младшего к старшему разряду.
5. Результатом работы преобразователя является шестнадцатиразрядное двоичное число (запись может содержать незначащие нули).

На вход преобразователю подали запись числа, равного $FEDCBA9876543210_{16}$ и получили некоторое число M . Сколько всего существует таких чисел, которые можно подать на вход преобразователю и получить в результате такое же число M ? Поскольку число может получиться достаточно большое, посчитайте и укажите в ответе сумму его цифр при записи в десятичной системе счисления.

4. Алгоритмизация и программирование. Формальный исполнитель (1 балл)

[Лабиринт]

Дан лабиринт, размером 10 на 10 клеток:

```
* . * * * * * * * *
* . . . . . . . . *
* . * . * * . * . *
* . * . . . . * . *
* . * . * * . * . *
* . * . * * . * . *
* . * . . . . * . *
* . * * * * * . *
* . . . . . . . . *
* * * * * * * * *
```

Каждая клетка содержит один из двух символов: «*» означает непреодолимое препятствие, а «.» свободная клетка. В одну из свободных клеток помещают шарик. Если ниже этой клетки есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. Далее лабиринт поворачивают следующим образом:

1. На 90 градусов по часовой стрелке
2. На 90 градусов по часовой стрелке
3. На 90 градусов по часовой стрелке
4. На 90 градусов против часовой стрелки.

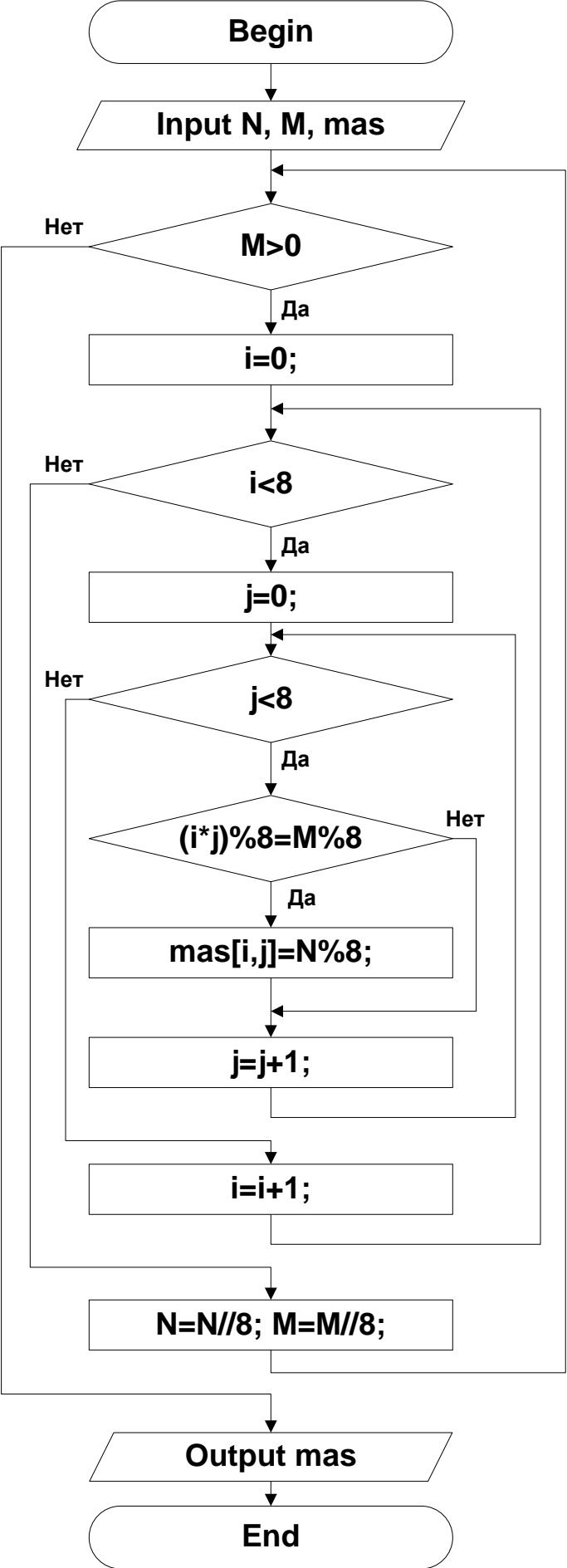
После каждого поворота, если ниже той клетки, в которой находится шарик, есть еще свободные клетки, шарик падает, пока не наталкивается на непреодолимое препятствие. После этого можно осуществлять следующий поворот.

Сколько существует клеток, в которые можно исходно поместить шарик так, что в результате выполнения указанных поворотов (возможно до их завершения), он выкатится из лабиринта? В ответе укажите целое число.

5. Алгоритмизация и программирование. Анализ алгоритма, заданного в виде блок-схемы (2 балла)

[Странная симметрия]

Дана блок-схема алгоритма.



На вход алгоритму подаются два целых положительных числа N и M и двумерный массив mas размером 8×8 элементов инициализированный нулями.

Известно, что на вход было подано число $M=8143989$. Определите минимальное значение числа N , такое, что на выходе будет массив mas со следующими значениями:

5	5	5	5	5	5	5	5
5	2	3	7	4	0	1	6
5	3	4	1	5	3	4	1
5	7	1	2	4	6	3	0
5	4	5	4	5	4	5	4
5	0	3	6	4	2	1	7
5	1	4	3	5	1	4	3
5	6	1	0	4	7	3	2

В ответе укажите целое число. Если такого числа не существует, запишите в ответ NULL.

Примечания:

1. При обращении к элементам массива первый индекс означает номер строки, а второй – номер столбца, нумерация строк и столбцов начинается с нуля.
2. Операция $A\%B$ означает вычисление остатка при целочисленном делении A на B , а операция $A//B$ означает вычисление частного при целочисленном делении A на B .

6. Телекоммуникационные технологии (2 балла).

[Wildcard Mask]

В IPv4 сетевой адрес представляет собой 4-байтное число, записанное в десятичной форме, где отдельные байты разделены точкой. В IP-адресе выделяются две части – адрес сети и адрес узла. Деление происходит с помощью маски – 4-х байтного числа, которое поставлено в соответствие IP-адресу. Маска содержит двоичные 1 в тех разрядах IP-адреса, которые определяют адрес сети и двоичные 0 в тех разрядах IP адреса, которые определяют адрес узла.

Кроме обычной маски существует еще wildcard mask. Эти маски часто применяются для написания правил фильтрации в межсетевых экранах (firewall) для того, чтобы определить применяется ли правило к IP адресу из проверяемого пакета или нет. Wildcard mask – 4-байтное число, которое содержит двоичные единицы в тех разрядах IP адреса, которые могут в IP адресе содержать разные значения, и нули в тех разрядах адреса, которые должны совпадать с заданными в правиле фильтрации. Wildcard mask не обязана содержать подряд идущие единицы или нули.

Межсетевой экран — программный или программно-аппаратный элемент компьютерной сети, осуществляющий контроль и фильтрацию проходящего через него сетевого трафика в соответствии с заданными правилами. В этой задаче мы будем использовать синтаксис команд для оборудования Cisco. Реализована отдельная процедура составления правил. Проверка правил идет сверху вниз по списку до первого срабатывания.

Рассмотрим пример, в котором используются обычные mask и wildcard mask. Будем считать, что для определения типа маски используется её старший бит: 1 – обычная маска, а 0 – wildcard mask. В примере опущены команды, необходимые для переключения контекстов управления и активации списка правил.

правило 1. создать список контроля с именем 123 и добавить в него правило, пропускающее пакеты с одного адреса 192.168.0.11 на любые адреса. То, что правило работает для одного адреса, видно из использованной маски, состоящей целиком из 32-х единиц.

```
access-list 123 permit ip 192.168.0.11 255.255.255.255 any
```

правило 2. добавить в список 123 правило, запрещающее передачу пакетов со всех нечетных адресов сети 192.168.0.0 255.255.255.0 по любым направлениям

```
access-list 123 deny ip 192.168.0.1 0.0.0.254 any
```

Рассмотрим подробнее правило с wildcard mask. Предположим, на порт поступает пакет с адреса 192.168.0.127. И надо проверить, применять ли правило 2 к этому адресу. Представим в двоичном формате проверяемый адрес, адрес-образец из правила и wildcard mask.

Ячейку B2 скопировали во все ячейки диапазона C2:I2. Ячейку B3 скопировали во все ячейки диапазона B3:H102. Ячейку I3 скопировали во все ячейки диапазона I4:I102. Диапазон A3:A102 заполнили некоторым набором целых положительных чисел.

Известны значения некоторых ячеек, получившиеся после этого:

	A	B	C	D	E	F	G	H	I	J
1		2	3	5	6	10	15	30		
2		50	33	20	17	10	7	4		
3										
4										
5										

Определите значение в ячейке I2. В ответе укажите целое число.

8. Технологии программирования (2 балла)

[Биржа]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

В данной задаче вам требуется реализовать подсчет прибыли при торговле акциями.

Торги проходят в течении n дней, всего на рынке представлены акции m компаний, цены акций компании фиксированы в течение одного дня. Сделки бывают двух типов:

Купит x акций компании $comp$

Продать все акции компании $comp$

За каждую сделку надо заплатить 1% комиссии. Например, если купить 10 акций по 300 рублей, то суммарно заплатить придется 3030 рублей. Если же продавать 10 акций стоимостью 300 рублей каждая, то за них можно получить 2970 рублей.

Прибылью с продажи будем считать разность полученных при продаже денег и суммарно потраченных денег при покупках. Например, если 10 акций были куплены по 300 рублей, а затем еще 5 акций были куплены по 400 рублей, то в случае продажи по стоимости 500 прибыль составит: $15 \cdot 500 \cdot 0.99 - (10 \cdot 300 \cdot 1.01 + 5 \cdot 400 \cdot 1.01) = 7425 - (3030 + 2020) = 2375$ рублей. При этом акции могут быть проданы в убыток (за меньшую стоимость, чем были куплены), тогда прибыль с продажи будем считать отрицательной.

Суммарной прибылью в момент времени будем считать сумму всех прибылей с продаж, которые произошли до этого момента. Обратите внимание, что деньги, потраченные к этому моменту на покупку акций, которые еще не были проданы, не учитываются. Также будем считать, что у нас неограниченный бюджет для покупки акций, но рассматривать будем только прибыль. Число акций для покупки/продажи для каждой компании на рынке также не ограничено.

Вам даны k событий покупки/продажи. Необходимо найти минимальную суммарную прибыль среди всех моментов времени.

Формат входных данных

В первой строке входных данных содержится одно целое число t — число тестовых наборов ($1 \leq t \leq 30$).

Затем следуют t тестовых наборов. Каждый тестовый набор описывается следующим образом:

В первой строке тестового набора содержатся три целых числа n , m и k — число дней, в которые проходят торги, число компаний на рынке и число событий, соответственно ($1 \leq n, m \leq 100$, $1 \leq k \leq 1000$).

В следующих m строках записаны названия компаний и n чисел — стоимости акций компании в рублях в каждый из дней торгов. Названия компаний состоят из не более чем 10 строчных букв латинского алфавита и попарно различны. Стоимости акций — целые числа в диапазоне от 1 до 10^5 включительно.

В следующих k строках заданы события покупки/продажи в хронологическом порядке. Событие покупки задается в формате **<день> buy <число акций> <название компании>**, а событие продажи задается в формате **<день> sell <название компании>**. При этом **<день>** — целое число от 1 до n , а **<число акций>** — целое число от 1 до 1000. Гарантируется, что все события следуют в порядке неубывания дней и корректны, а именно нет продаж некупленных акций и покупок акций, которых нет на рынке.

Формат выходных данных

Для каждого тестового набора выведите в отдельной строке минимальную прибыль среди всех моментов времени, с относительной или абсолютной погрешностью не более 10^{-4} .

Пример

Стандартный ввод	Стандартный вывод
4	0
3 1 3	-11.11
comp 300 400 500	-2080.0
1 buy 10 comp	0
2 buy 5 comp	
3 sell comp	
3 2 4	
gazp 100 111 300	
yndx 1000 1100 1111	
1 buy 10 gazp	
2 buy 1 yndx	
3 sell yndx	
3 sell gazp	
3 1 3	
comp 300 400 200	
1 buy 10 comp	
2 buy 5 comp	
3 sell comp	
2 2 3	
bdn 100 100	
nik 1 100	
1 buy 300 bdn	
1 buy 10 nik	
2 sell nik	

Замечание

В первом тестовом наборе изначально до продаж суммарная прибыль равна 0, после первой продаже суммарная прибыль становится 2375 (случай разобран в примере).

Во втором тестовом наборе промежуточные прибыли равны 0, -11.11 (акция продана дороже, но комиссия больше разницы) и 1948.89.

В третьем тестовом наборе промежуточные прибыли равны 0 и -2080.

В четвертом тестовом наборе промежуточные прибыли равны 0 и 979.9, деньги, потраченные на непроданные акции, не учитываются.

9. Технологии программирования (4 балла)

[Трансформация массива]

Имя входного файла	стандартный ввод
Имя выходного файла	стандартный вывод
Ограничение по времени	2 секунды
Ограничение по памяти	256 мегабайт

Вам дано t пар массивов a_i и b_i равной длины.

За одну операцию модификации можно:

Поменять местами любые два элемента массива a_i , но каждый элемент массива может участвовать не более чем в одном обмене.

Прибавить к любому элементу массива a_i единицу. Данную операцию можно применять неограниченное число раз к любому элементу массива.

Для каждой пары массивов найдите минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив b_i , или определите, что это невозможно.

Формат входных данных

В первой строке дано число t — число пар массивов ($1 \leq t \leq 40$).

В следующих $3t$ строках содержатся описания пар массивов. Каждая пара описывается тремя строками.

В первой из них дано число n_i — количество элементов в каждом массиве i -й пары ($1 \leq n_i \leq 10$). Во второй строке заданы n_i чисел $a_{i,j}$ — элементы массива a_i ($1 \leq a_{i,j} \leq 1000$). В третьей строке заданы n_i чисел $b_{i,j}$ — элементы массива b_i ($1 \leq b_{i,j} \leq 1000$).

Гарантируется, что сумма n_i по всем тестовым наборам не превосходит 150.

Формат выходных данных

Для каждой пары массивов выведите одно число — минимальное число операций, которые необходимо применить к массиву a_i , чтобы получить массив $b_{i,j}$, или -1, если для данной пары это невозможно.

Пример

Стандартный ввод	Стандартный вывод
6	-1
1	-1
2	1
1	-1
4	7
1 2 3 4	-1
2 4 1 2	
1	
1	
2	
5	
1 2 3 4 5	
1 5 1 1 1	
5	
1 2 3 4 5	
5 1 5 5 5	
3	
1 2 3	
3 1 2	