

I. Задания заключительного этапа олимпиады 2020-21 года

Заключительный этап 11 класса (приведен один из вариантов заданий)

1. Кодирование информации. Системы счисления (2 балла)

[Периодическая система]

Дана система уравнений, неизвестными в которой являются основания двух позиционных систем счисления X и Y :

$$\begin{cases} 0,(34)_X + 0,(13)_Y = 1 \\ 2 * 101_X - 102_Y = 1 \end{cases}$$

где левая часть первого уравнения представляет собой сумму двух периодических дробей.

Найдите и запишите в ответ через пробел сначала значение X , а затем значение Y , удовлетворяющие этой системе уравнений. Если таких пар несколько, запишите пару с наименьшим значением X . Если такой пары не существует, запишите в ответ NULL.

2. Кодирование информации. Объем информации (2 балла)

[ABC]

Петя решил построить генератор случайных последовательностей символов. Генератор состоит из трех источников символов, общего буфера и таймера. Таймер отсчитывает секунды, начиная с 1. Источник 1 каждую секунду записывает в буфер один символ, по кругу выбирая символы из набора {A, B, C}. Источник 2 каждую секунду, кратную 2, записывает в буфер один символ A. Источник 3 каждую секунду, кратную 3, записывает в буфер один символ, по кругу выбирая символы из набора {A, B}. Если в некоторую секунду несколько источников должны записать в буфер свои символы, они всегда делают это по порядку возрастания их номеров.

Вот результаты работы генератора на первых 6 секундах:

1. A
2. ABA
3. ABACA
4. ABACAAA
5. ABACAAAB
6. ABACAAABCA

Петя настолько уверен в надежности своего генератора, что, сформировав в буфере строку из N символов, решил записать её в память как последовательность двоичных кодов отдельных символов, используя для кода каждого символа одинаковое минимально возможное количество бит. Вася проанализировал работу генератора Пети и убедился, что он не такой уж надежный. Вася увидел, что длина строки Пети N кратна трем, и разбил строку в буфере на тройки идущих подряд символов, начиная с начала строки. Он посчитал, сколько всего различных троек символов встречается в строке, и решил записать в память строку в виде последовательности двоичных кодов троек символов, используя для кода каждой тройки одинаковое, минимально возможное количество бит. Вася обнаружил, что ему потребовалось для записи в память строки на 40 бит меньше, чем Пете. Определите и запишите в ответ длину строки N.

- - - -

3. Основы логики (1 балл)

[Эквиваленции]

Известно следующее равенство:

$$((\bar{B} \rightarrow A) \rightarrow \bar{B}) \vee (A \rightarrow C) \wedge C \vee A \rightarrow \bar{\bar{D}} = \text{ложь}$$

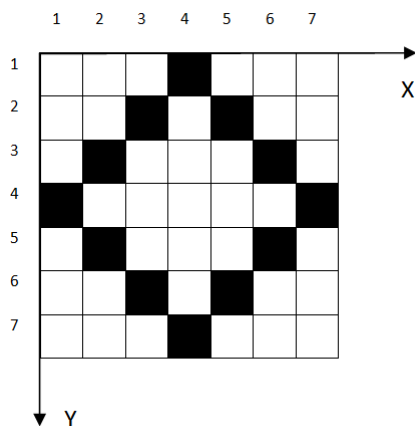
Тогда для каких из перечисленных логических выражений можно однозначно определить их логическое значение (истинность или ложность)? Отметьте **все** подходящие выражения.

1. $A \wedge B \leftrightarrow C \wedge D$
2. $(A \rightarrow B) \leftrightarrow (C \rightarrow D)$
3. $A \leftrightarrow B \wedge C \wedge D$
4. $A \wedge C \leftrightarrow B \wedge D$
5. $(A \rightarrow C) \leftrightarrow (B \rightarrow D)$
6. $B \leftrightarrow A \wedge C \wedge D$
7. $B \wedge C \leftrightarrow A \wedge D$
8. $C \leftrightarrow A \wedge B \wedge D$
9. $D \leftrightarrow A \wedge B \wedge C$
10. $(D \rightarrow C) \leftrightarrow (A \rightarrow B)$

4. Алгоритмизация и программирование. Формальные исполнители (3 балла)

[Рекурсивное закрашивание]

Петя очень давно участвует в олимпиадах и знает, что есть алгоритм рекурсивного закрашивания растрового изображения. У Пети есть множество черно-белых (bitmap) изображений, размером N на N пикселей, где N – нечетное число. На каждом изображении присутствует контур, имеющий форму ромба, центр которого совпадает с центром изображения, а вершины расположены в центрах первой и последней строки, и первого и последнего столбца изображения, соответственно. Линии имеют толщину в один пиксель. Вот пример такого изображения, размером, 7 на 7 пикселей:



Традиционно для компьютерной графики, система координат имеет начало в верхнем левому углу (начальный пиксель имеет координаты (1,1)) и направления осей как показано на рисунке.

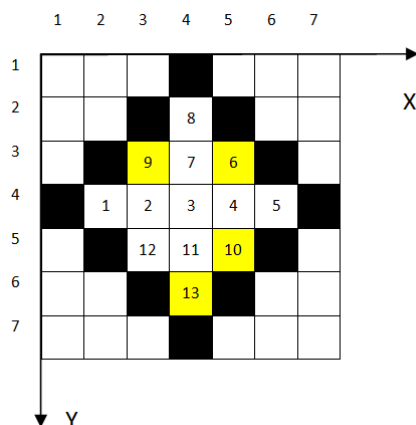
Алгоритм рекурсивного закрашивания заключается в рекурсивном вызове процедуры «Закрасить», которой передаются два параметра – координаты X и Y пикселя.

Процедура *Закрасить*(X,Y), может быть описана следующим образом:

1. Если цвет пикселя с координатами (X,Y) белый, то:
 - a. Изменить цвет пикселя с этими координатами на черный;
 - b. Вызвать процедуру *Закрасить*($X+1,Y$);
 - c. Вызвать процедуру *Закрасить*($X,Y-1$);
 - d. Вызвать процедуру *Закрасить*($X-1,Y$);
 - e. Вызвать процедуру *Закрасить*($X,Y+1$);
2. Иначе завершить процедуру.

Будем называть прыжком ситуацию, когда следующий закрашиваемый пиксель не находится в четырехсвязной окрестности предыдущего пикселя, то есть не является пикселем, прилегающим слева, справа, сверху или снизу к нему.

Петя решил закрашивать своё изображение, осуществив исходный вызов процедуры *Закрасить*(2,4) и проанализировал порядок, в котором закрашивались пиксели, выделив пиксели, закрашивание которых произошло в результате прыжков:

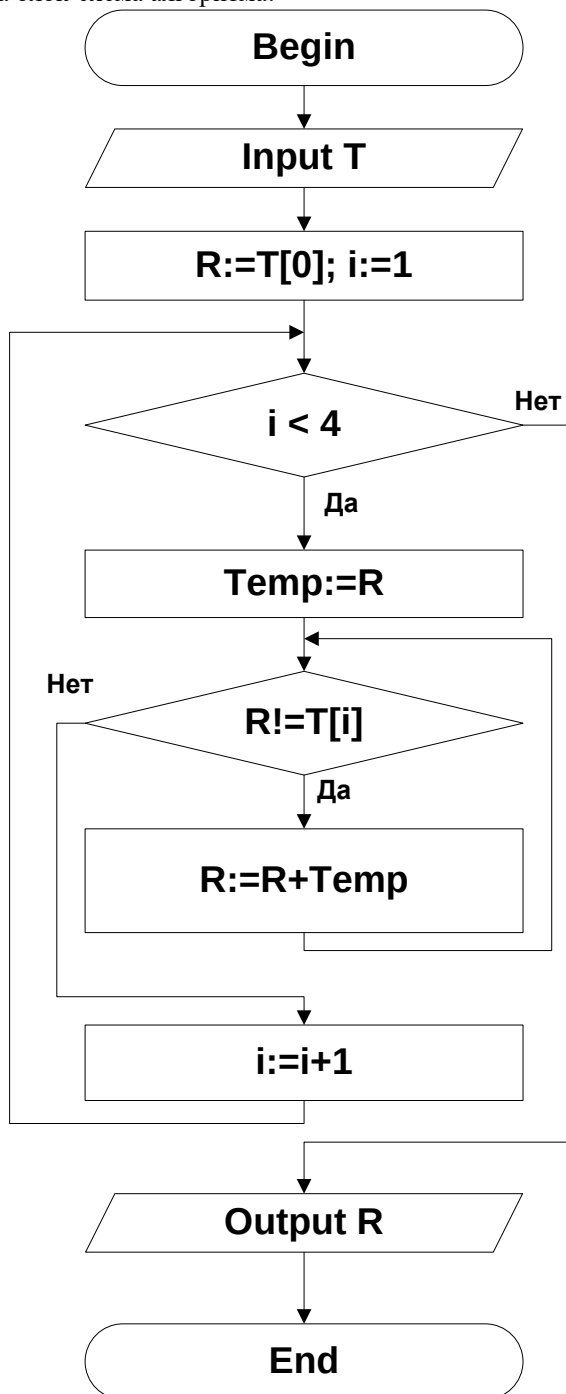


Пете стало интересно, и он решил узнать, сколько пикселей закрасится в результате прыжков, если взять изображение, размером 15 на 15 пикселей и осуществить исходный вызов процедуры закрашивания в пикселе, являющимся центром изображения. Определите их количество и запишите в ответ целое число.

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

[Пороги]

Дана блок-схема алгоритма:



На вход подается массив из четырех целых положительных чисел $T = [7, X, Y, Z]$. Все элементы массива попарно различны. Известно, что в результате выполнения алгоритма было выведено число 23023. Определите неизвестные значения элементов входного массива и укажите в ответе через пробел значения первых двух из них: сначала значение X и затем значение Y. Если существует несколько подходящих вариантов, укажите такой вариант, в котором сумма X и Y будет минимальна. Если такого массива не существует, укажите в ответе NULL.

Примечание: нумерация элементов массива идет с нуля, оператор « $\neq$ » означает «неравно».

6. Телекоммуникационные технологии (2 балла).

[EUI-64]

В протоколе IPv6 адрес представляет собой 128 битную двоичную последовательность, записываемую в шестнадцатеричном формате, группами по 4 цифры. Группы разделяются двоеточиями.

В адресе IPv6 есть две логические части:

1) первая часть, называемая префиксом, содержит адрес сети. По умолчанию она начинается с первого (начального) бита адреса и заканчивается на 64-м бите. Однако длина префикса может меняться. Длину префикса в битах указывают вместе с адресом, в виде десятичного числа, записанного через символ “/”

2) вторая часть, начинающаяся сразу после префикса – адрес узла (например, сетевой карты компьютера).

Для удобства IPv6 адрес можно записывать сокращенно для чего:

1) не пишутся ведущие нули в каждой группе

2) не указываются группы, содержащие только «0». Но это делается только для одной последовательности нулевых групп с конца адреса.

Пример:

для адреса

2001:0000:00AF:ABCD:0000:0000:0000:1234 / 64

сокращенная запись будет:

2001:0000:AF:ABCD::1234 / 64

В примере число бит «/64» показывает, что граница префикса проходит по середине адреса, а жирным выделены сокращенные при записи участки.

Часть IPv6 адреса, соответствующая адресу узла редко назначается вручную. Стандартным механизмом формирования адреса служит алгоритм EUI-64 при котором узлу сообщается префикс адреса, а вторую часть адреса узел генерирует сам на основании MAC адреса.

MAC адрес – это аппаратный адрес сетевого устройства, состоящий из 48 бит, записываемых в шестнадцатеричной форме. Адрес имеет свою структуру: первые 24 бита идентифицируют производителя, последние 24 бита назначаются производителем для конкретного устройства. MAC адреса не сокращаются.

MAC адрес, как правило может быть изменен пользователем, но, по стандарту, 7-й с начала адреса бит не может быть выбран произвольно. Действует правило:

1) Если адрес назначен на заводе, то 7-й бит равен 0,

2) если адрес установлен вручную, то 7-й бит равен 1.

Алгоритм EUI-64 сводится к следующим шагам:

1) MAC-адрес делится на две равные части;

2) Между частями вставляется комбинация "FFFE" (так что бы получилось 64-битное значение);

3) Инвертируется 7-й с начала бит получившейся комбинации.

Пусть в консоли маршрутизатора CISCO были набрана команда формирования IPv6 адреса с указанным префиксом по алгоритму EUI-64:

```
INT1(config-if)#ipv6 address 2001:1234:5678:abcd::/64 eui-64
```

После завершения работы команды была выполнена команда получения адресной информации по протоколу IPv6 и получен консольный вывод.

```
INT1#show ipv6 interface fa0/0
FastEthernet0/0 is up, line protocol is up
Global unicast address(es):
2001:1234:5678:ABCD:C000:FF:FE5C:1, subnet is
2001:1234:5678:ABCD::/64 [EUI]
```

Определите исходный MAC адрес конфигурируемого интерфейса и то, как был установлен адрес: установлен вручную или оставлен по умолчанию заводской адрес.

В ответ укажите MAC адрес в шестнадцатеричном виде и, через пробел букву L – в случае, если адрес назначен вручную и G, если адрес оставлен заводской адрес. Например, 001122334455 G.

7. Технологии обработки информации в электронных таблицах, сортировка и фильтрация данных (1 балл)

[5 чисел]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1		=СРЗНАЧ(A1:A3)	=МАКС(A1:A3)	
2				
3				
4				
5				
6				

В ячейках диапазона A1:A5 находятся пять неповторяющихся целых положительных чисел. Если выделить диапазон A1:A5 и отсортировать ячейки по убыванию, выяснится, что ячейка B1=18, а ячейка C1=25. Если же отсортировать ячейки этого диапазона по возрастанию, то ячейка B1=10, а ячейка C1=13. Известно, что в одной из ячеек содержится число 9. Определите значения всех ячеек диапазона A1:A5 и запишите в ответ в порядке убывания.

8. Технологии программирования (2 балла)

[Экспресс тест]

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

В данной задаче вам предлагается автоматизировать оценку результата экспресс теста.

Вам дана двухцветная картинка размером 10×20 . Для обозначения цветов используются символы «#» и «.». Тест считается отрицательным, если на картинке изображена одна вертикальная полоска и положительным, если три. В любом другом случае тест считается испорченным.

Полоской будем считать область картинки $10 \times k$, состоящую из символов «#», где k может быть произвольным. При этом все соседние клетки с этой областью должны быть «.». Полоска может находиться на границе картинки.

Формат входных данных

В первой строке входных данных задано число t - число тестов ($1 \leq t \leq 30$). В следующих $t \cdot 10 + (t - 1)$ строках заданы картинки тестов. Соседние картинки разделены пустыми строками. После последней картинки, пустой строки нет.

Каждая картинка состоит из 10 строк по 20 символов, каждый из которых либо «#», либо «.».

Формат выходных данных

Для каждой картинки выведите результат теста в отдельной строке:

Negative - если тест отрицательный;

Positive - если тест положительный;

Incorrect - если тест испорчен.

Пример

Стандартный ввод	Стандартный вывод
4##.....##.....##.....##.....##.....##.....##.....##.....##.....##.....##.....##..... ##.....##.....##.....#.....##.....##.....##.....##.....##.....##.....##..... ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..#.....#.....####. ..##.....#.....####. ..##.....#.....####. ..##.....#.....####. ..##.....#.....####. ..#.....#.....####. ..#.....#.....####. ..##.....#.....####. ..#.....#.....####.	Negative Incorrect Positive Incorrect

[Убрать циклы]

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

Мальчик Коля любит графы, но не любит циклы. У Коли есть несколько ориентированных не обязательно связанных графов. Графы могут содержать параллельные ребра, но не содержат петли. К сожалению, некоторые из них могут содержать циклы.

Коля хочет развернуть некоторые ребра в каждом графе, чтобы избавиться от циклов. Разворотом ребра будем считать замену ребра из вершины a в вершину b на противоположное ребро, которое будет направлено из вершины b в вершину a . При этом Коля хочет развернуть наименьшее число ребер в каждом графе, чтобы получившиеся графы не содержали циклов.

Помогите Коле узнать, какое наименьшее число ребер ему нужно развернуть в каждом графе.

Формат входных данных

В первой строке входных данных содержится число t - число графов ($1 \leq t \leq 8$).

В первой строке описания каждого графа дано два числа n и m - число вершин и число ребер соответственно ($2 \leq n \leq 10$, $1 \leq m \leq 1000$).

В следующих m строках содержатся пары чисел a_i, b_i - описание ребер графа ($1 \leq a_i, b_i \leq n, a_i \neq b_i$).

Для удобства, описания графов разделены пустыми строками. Пустой строки после последнего графа нет.

Гарантируется, что суммарное число вершин во всех графах не превосходит 50.

Формат выходных данных

Для каждого графа выведите одно число в отдельной строке - наименьшее число ребер, которые нужно развернуть, чтобы полученный граф не содержал циклов.

Пример

Стандартный ввод	Стандартный вывод
4	1
2 2	0
1 2	1
2 1	1
4 2	
1 4	
4 3	
3 3	
1 2	
2 3	
3 1	
3 5	
1 2	
2 3	
3 1	
2 3	
1 3	

Замечание

В первом графе нужно развернуть одно из ребер, чтобы убрать цикл. Во втором графе циклов изначально нет, так что ничего разворачивать не надо. В третьем графе можно развернуть любое ребро, чтобы убрать цикл. В четвертом графе нужно развернуть ребро (3,1) или ребро (1,2), чтобы новый граф не содержал циклов.