

# I. Задания заключительного этапа олимпиады 2016-17 года

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

### 1. Кодирование информации. Системы счисления (3 балла)

#### [Огромное число]

Вася получил длинную последовательность из цифр следующим образом. Он брал подряд натуральные числа, начиная с 1, переводил их в четверичную систему счисления и записывал результаты перевода друг за другом. Вот начало этой последовательности:

123101112132021222330313233100...

Вася остановился только тогда, когда дописал в конец последовательности четверичную запись числа  $1023_{10}$ .

Затем он представил, что это одно большое число, записанное в четверичной системе счисления, и перевел его в шестнадцатеричную систему счисления.

Определите, какая шестнадцатеричная цифра стоит в этом числе на 60-ой позиции, считая слева направо от начала числа, а затем какая шестнадцатеричная цифра стоит на 1000-ой позиции. В ответе укажите эти две шестнадцатеричные цифры в указанном порядке через пробел.

### 2. Кодирование информации. Объем информации (2 балла)

#### [Опорные кадры]

Вася разрабатывает прототип цифровой видеокамеры. Видеокамера снимает кадры размером 1920 пикселей по горизонтали и 1080 пикселей по вертикали. Цвета всех пикселей кодируются с использованием палитры из  $2^{24}$  оттенков.

Изначально Вася предполагал, что будет записывать в память каждый кадр независимо, как последовательность кодов цветов пикселей, используя для записи каждого кода минимально возможное одинаковое для всех пикселей количество бит. Никакой другой информации, кроме кодов цветов пикселей в память не записывается.

Затем Вася стал анализировать результаты и обнаружил, что если разбить последовательность кадров на идущие подряд наборы из 6 кадров, то относительно каждого такого набора справедлива следующая закономерность. Во втором кадре набора оказывается ровно 1 процент пикселей, цвета которых отличаются от цветов пикселей, находящихся на этих же позициях в первом кадре набора. В третьем кадре набора оказывается ровно 2 процента пикселей, цвета которых отличаются от цветов пикселей, находящихся на этих же позициях в первом кадре набора и так далее. То есть, в каждом последующем кадре набора оказывается на 1 процент больше чем в предыдущем кадре набора пикселей, цвета которых отличаются от цветов пикселей, находящихся на этих же позициях в первом кадре набора.

Тогда Вася решил использовать более сложный способ записи данных в память. Для каждого набора он стал сначала записывать опорный кадр – первый кадр набора – как последовательность кодов цветов всех пикселей этого кадра, используя для записи каждого кода минимально возможное одинаковое для всех пикселей количество бит. А затем для каждого из оставшихся 5-ти кадров набора он стал записывать в память только разницу с опорным кадром. Для этого он закодировал все позиции пикселей в кадре и для каждого кадра набора, кроме опорного, стал записывать множество пар значений: код позиции пикселя, цвет которого изменился, по отношению к цвету пикселя на этой же позиции в опорном кадре и код нового цвета этого пикселя. Для записи каждого кода позиции пикселя Вася использует минимально возможное одинаковое для всех кодов позиций количество бит. Для записи кода цвета Вася использует такое же количество бит, как и для записи кода цвета в опорном кадре. Все получаемые коды Вася записывает подряд побитно.

Определите, какой выигрыш по памяти получает Вася при новом способе записи данных в памяти для одного набора из 6 кадров. Округлите этот выигрыш в меньшую сторону до целого числа МБайт и запишите в ответ.

Примечание 1МБайт=1024 КБайт, 1 КБайт=1024 байт.

### 3. Основы логики (1 балл)

#### [Таблица истинности]

Дан фрагмент таблицы истинности логической функции  $F(A,B,C,D)$ , зависящей от четырех аргументов A, B, C и D.

A	B	C	D	F
0	0	1	1	0
1	0	0	1	1
1	1	0	1	0

Известно, что эту функцию можно задать в виде следующего логического выражения:

$(([\dots] A \rightarrow [\dots] B) \rightarrow [\dots] C) \rightarrow [\dots] D$ ,

где вместо некоторых [...] может быть подставлен оператор логического отрицания. Определите, перед какими аргументами должны стоять операторы логического отрицания, чтобы получившаяся функция соответствовала приведенной таблице истинности.

В качестве ответа приведите последовательность из четырех знаков «+» или «-» в которой знак «+» будет означать, что, перед соответствующим аргументом в выражение не будет стоять оператор логического отрицания, а знак «-» будет означать, что перед соответствующим аргументом в выражение будет стоять оператор логического отрицания. Например, ответ «++-+» будет соответствовать выражению  $((A \rightarrow B) \rightarrow \text{not } C) \rightarrow D$

#### 4. Кодирование информации. Алгоритмы обработки кодированной информации (1 балл)

##### [Обратная польская нотация]

Обратная польская нотация (ОПН) – используемая в некоторых языках программирования форма записи математических выражений, в которой операнды расположены перед знаками операций.

Запись и вычисление выражения в ОПН устроены следующим образом:

- Выражение является последовательностью операндов и знаков операций.
- Выражение читается слева направо.
- Если в выражении встречается операнд, он заносится в стек операндов.
- Если в выражении встречается знак операции, выполняется соответствующая операция над двумя последними операндами в стеке и эти два операнда заменяются на результат вычисления операции.
- Результатом вычисления выражения становится результат последней вычисленной операции.

Пример вычисления выражения:  $7\ 2\ 3\ +\ -\ 4\ *$

Шаг	Текущее выражение	Стек операндов
1	$7\ 2\ 3\ +\ -\ 4\ *$	7
2	$2\ 3\ +\ -\ 4\ *$	7 2
3	$3\ +\ -\ 4\ *$	7 2 3
4	$\ +\ -\ 4\ *$	7 5
5	$\ -\ 4\ *$	2
6	$\ 4\ *$	2 4
7	$\ *$	8

Дано выражение, записанное в ОПН:

2 4 4 #1 #2 8 1 #1 #3

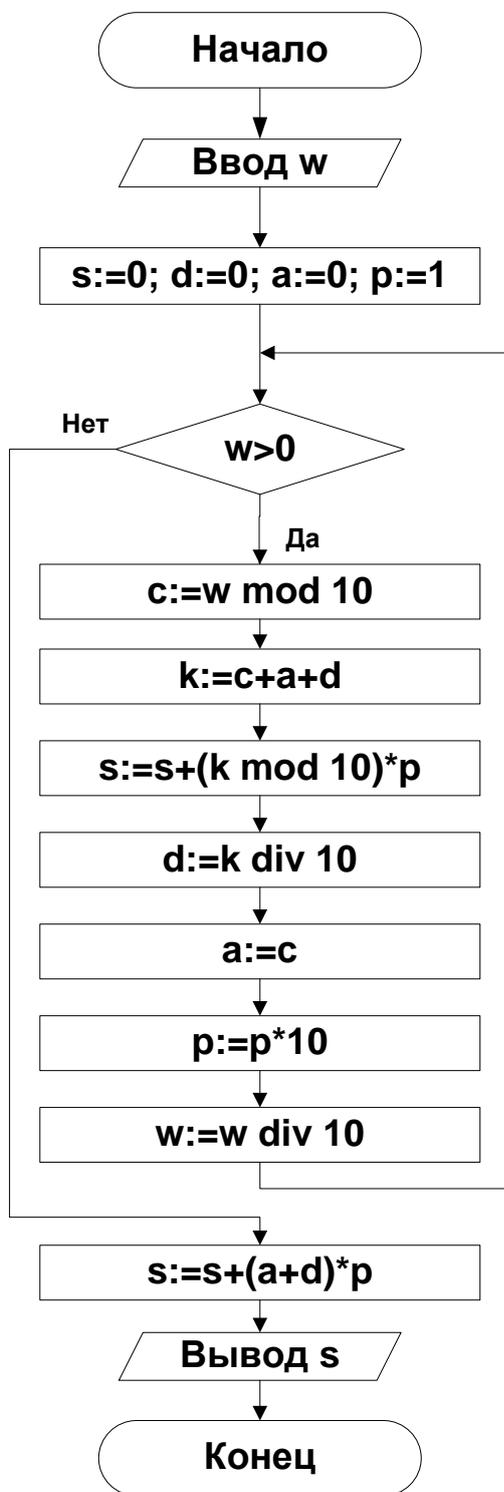
За символами “#1”, “#2” и “#3” скрываются операции сложения, вычитания и умножения (в выражении присутствуют все три операции), но неизвестно, какая операция соответствует какому символу. Подберите такой вариант соответствия символов операциям, при котором значение вычисленного выражения будет максимальным, и вычислите это значение. В ответе укажите целое число.

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

##### [Вычислитель]

Дана блок-схема алгоритма. Какое целое положительное число  $w$  необходимо подать на вход, чтобы после завершения алгоритма получилось значение  $s=96415$ ? В ответе укажите целое число.

*Примечание. Операция  $\text{mod}$  вычисляет остаток от деления первого аргумента на второй. Операция  $\text{div}$  вычисляет частное от целочисленного деления первого аргумента на второй.*



## 6. Алгоритмизация и программирование. Формальные исполнители (2 балла)

### [Петляющий робот]

Робот движется по плоскости, с заданной на ней прямоугольной декартовой системой координат (ось X направлена слева направо, ось Y направлена снизу вверх), следующим образом:

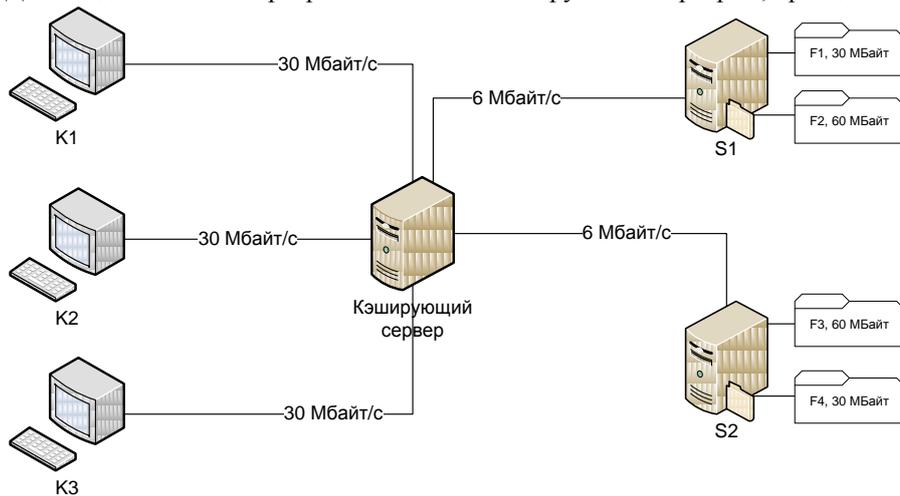
1. Робот может двигаться вперед или поворачиваться на 90 градусов против часовой стрелки.
2. Каждый ход робота – это изменение одной координаты на 1 в соответствии с текущим направлением его движения.
3. В памяти робота хранятся две переменных: A и B. После каждого хода робот увеличивает переменную B на 1 и делает проверку  $A=B$ .
4. Если проверка дает положительный результат, то робот выполняет следующие действия:
  - a. повернуться на 90 градусов против часовой стрелки;
  - b. обнулить переменную B.
  - c. уменьшить переменную A на 2. Если после этого она стала равна 0, то записать в переменную A значение 8.
5. Описанные в пункте 4 действия не считаются отдельным ходом.

Робот начал движение вправо из точки с координатами (0,0). В момент начала движения переменная  $A=8$ , а переменная  $B=0$ . Робот остановился, сделав 1000 ходов. Определите, сколько существует точек на плоскости, в которых робот был два раза и координаты последней такой точки. В ответе укажите через пробел три целых числа: сначала количество точек на плоскости, в которых робот побывал два раза, затем координату X последней по пути следования робота точки, в которой он побывал два раза, а затем координату Y этой точки.

## 7. Телекоммуникационные технологии (2 балла)

### [Кэширующий сервер]

Дана модель клиент-серверной системы с кэширующим сервером, приведенная на схеме:



Модель работает по следующим принципам.

1. На двух серверах данных S1 и S2 хранятся 4 файла с данными (F1 и F2 на сервере S1, F3 и F4 на сервере S2). Файлы F1 и F4 имеют размер по 30 МБайт каждый, а файлы F2 и F3 – по 60 МБайт каждый. Сервера данных связаны с кэширующим сервером каналами передачи данных со скоростями 6 МБайт/с. Кэширующий сервер связан с тремя клиентами каналами передачи данных со скоростями 30 МБайт/с.
2. Три клиента должны скачать все 4 файла каждый. У каждого клиента есть свой план, определяющий последовательность скачивания файлов. Клиенты в соответствии со своими планами присылают запросы на кэширующий сервер для скачивания конкретного файла. Клиент посылает запрос для скачивания следующего файла в момент, когда закончил скачивание предыдущего. Кэширующий сервер получает запрос мгновенно и сразу же начинает передачу данных клиенту из своей памяти или с сервера данных в соответствии с правилами, указанными ниже. Любой сервер может одновременно передавать данные нескольким клиентам.
3. Если в памяти кэширующего сервера **в момент поступления запроса** нет целиком нужного файла (в том числе, если он сейчас кэшируется, но кэширование не завершено), файл будет целиком скачан с сервера данных, на котором хранится этот файл. В этом случае скорость передачи данных клиенту равна скорости передачи данных в канале между сервером данных и кэширующим сервером. Скорость передачи данных между клиентом и кэширующим сервером в этом случае не учитывается.
4. Если передача некоторого файла запрашивается в первый раз, то кэширующий сервер начинает параллельно с передачей этого файла клиенту кэшировать его в свою память. Кэширование происходит с той же скоростью, что и передача файла клиенту и завершается в момент завершения передачи этого файла клиенту. Одновременные процессы кэширования нескольких файлов никак не влияют друг на друга. Процесс кэширования никак не влияет на процессы скачивания файлов. Кэширующий сервер имеет достаточно памяти для хранения всех четырех файлов и после кэширования хранит их сколь угодно долго.
5. Если в памяти кэширующего сервера **в момент поступления запроса** находится нужный файл, то он будет скачан с кэширующего сервера. Скорость скачивания клиентом файла с кэширующего сервера постоянная, зависит только от скорости передачи данных в канале между клиентом и кэширующим сервером и не зависит от количества одновременно выполняющихся операций скачивания данных клиентами или кэширования других файлов.
6. Если одновременно происходит передача нескольких файлов с одного сервера данных клиентам или передача одного и того же файла с сервера данных нескольким клиентам, то канал между сервером данных и кэширующим сервером на это время делится поровну на количество одновременных потоков передачи данных. Это значит, что скорость передачи уменьшается пропорционально количеству одновременных потоков получения данных. Даже если передается один и тот же файл одновременно нескольким клиентам, передача происходит независимыми потоками с разделением канала передачи данных.

Известно, что у клиентов были следующие планы скачивания файлов:

Клиент	Первый файл	Второй файл	Третий файл	Четвертый файл
K1	F1	F2	F3	F4
K2	F3	F4	F1	F2
K3	F1	F3	F2	F4

Все клиенты начали одновременно скачивать файлы в соответствии со своими планами. Определите время, через которое у всех клиентов будут скачаны все файлы. В ответе укажите целое число секунд.

## 8. Технологии обработки информации в электронных таблицах (1 балл)

### [Остатки]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E
1					
2	0	=ОСТАТ(\$A2;B\$1)		=СУММ(B2:C2)	
3	1				
4	2				
5	3				
6	4				
7	5				
8	6				
9	7				
10	8				

Диапазон ячеек A2:A100 заполнили последовательно по возрастанию целыми числами, начиная с 0. Ячейку B2 скопировали во все ячейки диапазона B2:C100. Ячейку D2 скопировали во все ячейки диапазона D3:D100. В ячейку B1 поместили число 15. Какое **минимальное** целое положительное число необходимо поместить в ячейку C1, чтобы в столбце D значение ячейки равно 1 первый раз (считая сверху вниз) встретилось в строке со значением ячейки в столбце A равно 45? В ответе укажите целое число.

*Примечание. В англоязычной версии Microsoft Excel и в OpenOffice функции ОСТАТ соответствует функция MOD, а функции СУММ – функция SUM.*

## 9. Технологии сортировки и фильтрации данных (3 балла)

### [Поисковые запросы]

Даны 9 запросов к поисковому сервису некоторого сегмента сети Интернет:

1. диск & видеокарта | процессор
2. процессор & диск & видеокарта
3. процессор | память & видеокарта
4. процессор & память & диск & видеокарта
5. процессор & диск | память & процессор
6. процессор | память | диск
7. (память | диск) & процессор & видеокарта
8. видеокарта | процессор
9. видеокарта & память & процессор

В каждом запросе использованы ключевые слова и логические операции: “И” (обозначена “&”) и “ИЛИ” (обозначена “|”), приоритет логической операции “И” выше, чем приоритет логической операции “ИЛИ”. Во время выполнения всех 9 запросов сегмент сети оставался неизменным. В результате выполнения каждого запроса может быть выдано некоторое количество соответствующих ему страниц.

Требуется выбрать из предложенных ниже вариантов все те, которые соответствуют расположению номеров запросов в порядке невозрастания количества страниц, соответствующих запросу. Будем считать, что если для некоторой пары запросов невозможно однозначно определить порядок невозрастания количества страниц, соответствующих этим запросам, то они могут между собой следовать в любом порядке.

Варианты ответов:

1. 861357294
2. 681392574
3. 683152974
4. 631857294
5. 683157924
6. 681357924
7. 863175294
8. 683157294

Для доступа к ответам нажмите «Ответить».

## 10. Технологии программирования (2 балла)

В Берляндии каждый автомобиль имеет регистрационный номер. Автомобильные номера в Берляндии имеют следующий вид: **LDDLDDL**, где символ **L** обозначает строчную латинскую букву, а **D** цифру.

Филипп устроился работать в службу регистрации автомобильных номеров. По своей неопытности в первый же день работы Филипп разлил на стопку номеров кофе. У некоторых номеров оказался залит второй блок цифр (цифры на позициях 5 и 6).

Филипп считает, что все номера в Берляндии уникальны, поэтому он хочет быстро подобрать все залитые цифры, так чтобы среди всех номеров не было двух одинаковых. Задача показалась ему нерешаемой, и он попросил вас помочь ему.

### Формат входного файла

В первой строке входного файла **input.txt** записано натуральное число **n**, не превосходящее **1000** — количество номеров в стопке.

В следующих **n** строках находятся **n** регистрационных номеров, в **i+1**-й строке **i**-й номер, в описанном выше формате. На месте залитых цифр находятся знаки вопросов.

Гарантируется, что знаки вопроса могут находиться только на месте цифр из второго блока цифр, причем либо на позициях обеих цифр, либо ни на одной из позиций.

### Формат выходного файла

Первая строка выходного файла **output.txt** должна содержать **NO**, если в стопке были одинаковые номера. Иначе первая строка должна содержать **YES**, а далее **n** строк должны содержать номера из стопки — по одному в каждой строке, причем **i+1**-я строка должна содержать **i**-й номер. Номера должны удовлетворять принятому в Берляндии формату в том же порядке, что и во входном файле.

Если ответов несколько — разрешается вывести любой.

Обратите внимание, что Филипп хочет восстановить только залитые цифры, то есть он должен заменить только знаки вопросов цифрами, другие символы в номерах измениться не должны.

### Пример входных и выходных данных

input.txt	output.txt
4 a10a10c a30b??c a30b??c x70r??r	YES a10a10c a30b10c a30b22c x70r37r
3 a00b10c a00b10c c02y03x	NO
2 a99a??b a99a??b	YES a99a11b a99a22b

## 11. Технологии программирования (4 балла)

На уроке информатики учитель рассказал Васе про новый вид строк — *максимально-символьные* строки. Строка называется *максимально-символьной*, если символ, который встречается в ней максимальное количество раз, единственен. Например, строка **"abacaba"** — *максимально-символьная*, потому что единственный символ, который встречается максимальное количество раз в ней — **'a'**. В то же время строка **"cabacbac"** — не *максимально-символьная*, потому что символы **'a'** и **'c'** встречаются в ней максимальное количество раз, то есть не являются единственными.

После урока Вася сразу начал думать над следующей задачей: из данного набора символов составить как можно меньше *максимально-символьных* строк, используя все символы из набора ровно по одному разу в любом порядке. Вася не смог придумать решение этой задачи, поэтому обратился за помощью к вам. Помогите ему!

### Формат входного файла

В единственной строке входного файла **input.txt** записана строка **s**, характеризующая набор символов. Ее длина не превосходит **100**.

### Формат выходного файла

В первой строке выходного файла **output.txt** требуется вывести минимальное количество *максимально-символьных* строк **k**, которое можно составить из данного набора, используя каждый символ ровно один раз.

В следующих **k** строках выходного файла требуется вывести *максимально-символьные* строки составленные из данного набора.

Если существует несколько правильных ответов, разрешается вывести любой из них.

### Пример входных и выходных данных

input.txt	output.txt
abacaba	1 abacaba
abcabc	2 aab ccb
abc	3 a b c
cabacbac	2 bcb acasa