

Задания для 11 класса

Заключительный этап

1. Информация и её кодирование (1 балл)

Сломанный сумматор.

Петя собрал сумматор, работающий следующим образом: в три входных восьмиразрядных регистра заносятся три числа, переведенные в двоичную систему счисления, а в выходном регистре получается их сумма (количество разрядов в выходном регистре достаточно для записи любой возможной суммы чисел, помещенных во входные регистры). К сожалению, Петя допустил ошибку, и во всех трех входных регистрах один разряд с одним и тем же номером работает некорректно – либо всегда равен нулю, либо всегда равен единице. В результате, когда Петя поместил в первый входной регистр двоичную запись числа 148_{10} , во второй – 68_{10} , а в третий – 201_{10} , в выходном регистре получилась сумма равная 401_{10} . Определите, какой разряд работает некорректно во всех входных регистрах, и какое значение он всегда содержит.

В ответе укажите через пробел два числа – первое, номер этого разряда, считая **справа налево** от младшего разряда (младший разряд имеет номер 1), а затем 0 или 1 в зависимости от того, чему всегда равен этот разряд. Например, ответ "1 0" будет обозначать, что во входных регистрах крайний правый (младший) разряд всегда равен нулю.

Ответ: 

2. Информация и её кодирование (3 балла)

Алгоритм RLE

Кодирование длин серий (Run-length encoding, RLE) - алгоритм сжатия данных, который оперирует сериями данных, то есть последовательностями, в которых один и тот же символ встречается несколько раз подряд. При кодировании подстрока одинаковых символов, составляющих серию, заменяется строкой, которая содержит сам повторяющийся символ и количество его повторов. Таким образом, последовательность символов преобразуется в последовательность пар X и Y , где X - количество повторений символа Y . Даже если в последовательности символов указан один неповторяющийся символ, он все равно преобразуется в пару $1Y$. Минимальное количество бит, которые потребуются для хранения X и Y , зависит от максимального количества символов в кодируемых строках и мощности алфавита символов, которые могут встречаться в строках соответственно.

Например, если мы знаем, что максимальная длина строки 32 символа и алфавит состоит из четырех символов, то последовательность

AAAABBCDDD

после кодирования приобретет вид

4A2B1C3D

и потребует для хранения 28 бит.

Пусть по указанному алгоритму кодируются строки длиной ровно 8 символов из восьмисимвольного алфавита (A, B, C, D, E, F, G и H) и для хранения X и Y выбирается минимальное количество бит.

Сколько существует вариантов таких строк, отличающихся хотя бы одним символом, чтобы для хранения закодированного варианта каждой такой строки потребовалось ровно 12 бит.

Ответ: 

3. Основы логики (2 балла)

Сложная зависимость

Даны два логических выражения:

$X(A,B,C) = A \text{ and not } B \text{ or } C$

$Y(A,B,C) = \text{not } A \text{ and } C \text{ or } B$

Найдите логическую функцию $F(X,Y)$, такую, что если вместо X и Y подставить заданные выше логические выражения, то окажется, что $F(X(A,B,C), Y(A,B,C)) = B \text{ and not } C$. Если таких функций несколько – запишите любую из них.

В ответе запишите формулу, которая может содержать логические переменные X и Y и не более чем три логические операции. Если таких функций не существует, запишите в ответ NULL.

Комментарий по вводу ответа: операнды вводятся большими латинскими буквами; логические операции обозначаются, соответственно как **not**, **and** и **or**. Запись не должна содержать скобок.

Пример записи ответа: $X \text{ or not } Y$

Ответ: 

4. Алгоритмизация и программирование (2 балла)

Растущая последовательность

Дана *исходная* последовательность цифр:

123

Результирующая последовательность строится по следующему алгоритму:

На каждом шаге алгоритма берется копия *исходной* последовательности. В ней каждая цифра заменяется на N таких же цифр, где N – номер шага алгоритма, и результат дописывается в конец последовательности, полученной на предыдущем шаге. Результатом выполнения первого шага является просто исходная последовательность цифр.

Алгоритм завершает работу, если после очередного шага количество цифр в последовательности превышает 1000.

Например, после третьего шага алгоритма получится последовательность

123112233111222333

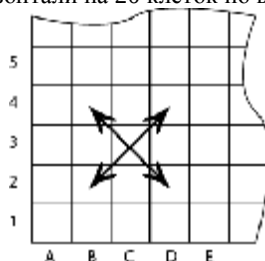
Сколько раз встретится последовательность цифр «12» в первых 1000 цифрах от начала последовательности?

Ответ: 

5. Алгоритмизация и программирование (2 балла)

Диагонали

Клетчатое поле, размером 18 клеток по горизонтали на 20 клеток по вертикали ограничено стенкой.



Робот передвигается по этому полю следующим образом:

1. За один шаг робот может сместиться по диагонали на одну клетку. Например, из клетки C3 робот может пойти в направлении «вправо-вверх» и оказаться в клетке D4, может «вправо-вниз» и оказаться в клетке D2, может «влево-вверх» и оказаться в клетке B4 и может «влево-вниз» и оказаться в клетке B2.
2. Робот сохраняет направление своего движения до тех пор, пока после совершения очередного шага он не оказывается в клетке, одна из границ которой является стеной, ограничивающей поле. Попад в эту клетку, он изменяет направление своего движения, развернувшись на 90 градусов по или против часовой стрелки и следующий шаг сделает уже в этом направлении. Например, если робот пришел в клетку C1 из клетки B2, то на следующем шаге он перейдет в клетку D2, а если он пришел в клетку C1 из клетки D2, то на следующем шаге он перейдет в клетку B2.
3. Если робот попадает в угловую клетку, то алгоритм завершается.

На каком шаге робот окажется в угловой клетке, если он начал свое движение из клетки A2 в направлении "вправо-вверх"? В ответе укажите целое число.

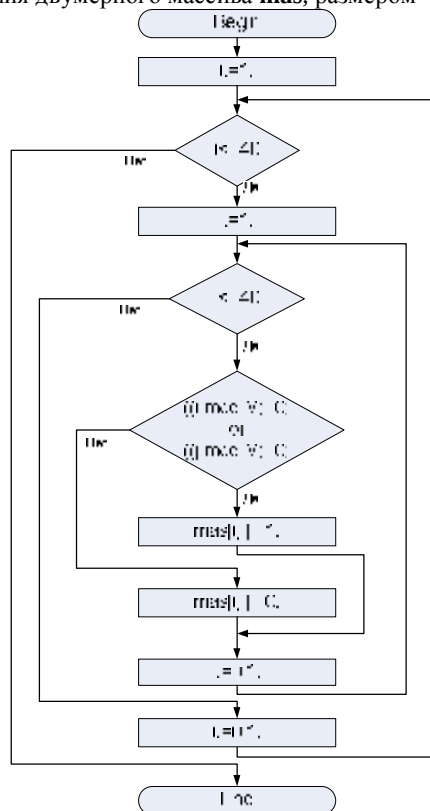
Примечание. Для нумерации клеток по горизонтали используются буквы латинского алфавита: A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R.

Ответ: ~~10~~

6. Алгоритмизация и программирование (2 балла)

Решётки

Дана блок-схема алгоритма заполнения двумерного массива **mas**, размером 40 на 40 элементов:



С использованием этого алгоритма были заполнены два массива – **masA** и **masB**, причем перед выполнением алгоритма для заполнения массива **masA** переменной **M** было присвоено значение 4, а перед выполнением алгоритма для заполнения массива **masB** переменной **M** было присвоено значение 6. Сколько получилось в массиве **masA** элементов, равных единице, таких, что в массиве **masB** элемент с таким же индексом также равен единице? В ответе укажите целое число.

Ответ: ~~10~~

7. Кодирование графической и звуковой информации (1 балл)

Aspect ratio

Дан перечень типовых разрешений экранов с указанием количества пикселей по длинной стороне. Известно, сколько КБайт памяти будут занимать четыре несжатых изображения такого разрешения, кодированные с использованием 24-хбитной цветовой палитры RGB. Найдите в этом списке те типовые разрешения, которые имеют соотношение сторон 3:4. В ответе укажите через пробел в порядке возрастания их номера.

Номер	Разрешение экрана	Количество пикселей по длинной стороне	Объем памяти, занимаемый четырьмя изображениями, КБайт
1	WQVGA	400	1125
2	VGA	640	3600
3	nHD	640	2700
4	SVGA	800	5625
5	WVGA	800	4500
6	qHD	960	6075
7	XGA	1024	9216
8	HD720P	1280	10800
9	WXGA	1280	11520
10	UXGA	1600	22500
11	FullHD	1920	24300

Отв

8. Телекоммуникационные технологии, фильтрация данных (3 балла)

Настройка брандмауэра

В стеке протоколов TCP/IP (версия 4) протокол IP осуществляет передачу сообщений между компьютерами. В заголовке сообщения протокола IP компьютер-отправитель идентифицируется IP-адресом отправителя, а компьютер-получатель - IP-адресом получателя. Внутри сообщения протокола IP помещается сообщение протоколов транспортного уровня (TCP или UDP), которые в свою очередь осуществляют передачу данных между отдельными программами, запущенными на этих компьютерах, причем в качестве адреса программы на каждом компьютере используется двухбайтный номер порта (для протоколов TCP и UDP порты нумеруются отдельно).

Одним из простейших способов обеспечения безопасности в IP-сетях является фильтрация сетевых сообщений формальным фаерволом (брандмауэром). Будем считать, что упрощенно принцип работы формального фаервола сводится к следующему:

- 1) фильтрация сетевого сообщения осуществляется по правилам;
- 2) правила выстраиваются в последовательности-цепочки и проверяются последовательно от начала к концу цепочки до первого срабатывания;
- 3) работает ли правило для сообщения или нет, определяется по комбинации формальных признаков: IP-адрес отправителя, IP-адрес получателя, порт отправителя, порт получателя. Для формирования правила может использоваться 1, 2, 3 или все 4 признака;
- 4) если анализируемое сообщение соответствует условиям правила, то для сообщения выполняется заданное действие (Пропустить или Заблокировать). Это означает, что правило сработало;
- 5) если правило сработало, то все последующие правила в цепочке для анализируемого пакета не выполняются.

Например, по правилу

«Если (адрес_отправителя=192.168.0.4 и порт_получателя_TCP=80) то Пропустить»

сообщение «адрес_отправителя=192.168.0.4, адрес_получателя=172.21.24.4 (порт_отправителя_TCP=34004, порт_получателя_TCP=80)» будет пропущено, а сообщение «адрес_отправителя=192.168.0.4, адрес_получателя=172.21.24.4 (порт_отправителя_TCP=80, порт_получателя_TCP=8080)» – нет.

Дан набор правил:

- 1) Если (адрес_отправителя=192.168.7.10 и порт_получателя_TCP=53) то Заблокировать
 - 2) Если (адрес_отправителя=192.168.7.10) то Заблокировать
 - 3) Если (порт_отправителя_TCP=5037) то Пропустить
 - 4) Если (порт_получателя_UDP=53) то Заблокировать
 - 5) Если (адрес_отправителя=180.37.4.19) то Заблокировать
 - 6) Если (порт_получателя_TCP=53 и порт_отправителя_TCP=5037) то Пропустить
 - 7) Если (адрес_получателя=10.15.25.35) то Пропустить
 - 8) Если (адрес_отправителя=172.21.13.7 и порт_получателя_TCP=53) то Пропустить
- Нужно, чтобы для перечисленных ниже пакетов выполнялись указанные действия:
- 1) «адрес_отправителя=192.168.7.10, адрес_получателя=10.10.9.1 (порт_отправителя_TCP=10080, порт_получателя_TCP=53)» – будет заблокирован
 - 2) «адрес_отправителя=192.168.7.10, адрес_получателя=10.10.9.2 (порт_отправителя_UDP=20150, порт_получателя_UDP=53)» – будет заблокирован
 - 3) «адрес_отправителя=92.122.7.16, адрес_получателя=10.15.25.35 (порт_отправителя_TCP=5037, порт_получателя_TCP=5773)» – будет пропущен
 - 4) «адрес_отправителя=92.122.7.10, адрес_получателя=172.21.13.7 (порт_отправителя_TCP=5037,

- порт_получателя_TCP=53)» – будет пропущен
 5) «адрес_отправителя=180.37.4.19, адрес_получателя=11.10.9.8 (порт_отправителя_TCP=5037, порт_получателя_TCP=783)» – будет заблокирован

Сформируйте из перечисленных выше правил цепочку минимальной длины, такую чтобы для приведенных пакетов в результате обработки по этой цепочке гарантировано выполнялись указанные действия (для каждого пакета должно сработать правило из цепочки и привести к выполнению требуемого действия).

В ответе укажите через запятую номера правил, начиная от начала цепочки к ее концу. Если вариантов таких цепочек несколько – укажите любой из них.

Ответ: _____

9. Технологии обработки информации в электронных таблицах (1 балл).

Дроби

Фрагмент электронной таблицы представлен в режиме отображения формул. Ячейку B1 скопировали в каждую ячейку диапазона B2:B6. Ячейку C1 скопировали в каждую ячейку диапазона C2:C6. В ячейку D1 поместили положительное вещественное число, меньшее единицы, все цифры которого не превосходят 4. Найдите это число, если известно, что в результате вычислений в ячейке C7 получилось значение 0,31104? В ответе укажите вещественное число, используя в качестве разделителя целой и дробной части запятую.

	A	B	C	D	E
1	1	=ОСТАТ(ОКРВНИЗ(D\$1*СТЕПЕНЬ(10;A1);1);10)	=ЕСЛИ(B1<>0;B1*СТЕПЕНЬ(E\$1;-A1);0)		5
2	2				
3	3				
4	4				
5	5				
6	6				
7			=СУММ(C1:C6)		
8					

Примечание:

Соответствие названий функций в различных редакторах электронных таблиц

Microsoft Excel (Rus)	Microsoft Excel (Eng)	OpenOffice.org Calc
ЕСЛИ	IF	IF
ОКРВНИЗ	FLOOR	FLOOR
ОСТАТ	MOD	MOD
СТЕПЕНЬ	POWER	POWER
СУММ	SUM	SUM

Ответ: _____

10. Технологии программирования (3 балла).

После проведения спортивных соревнований, например, по футболу, и организаторы, и участники, и зрители особенно интересуются различной статистикой. Количество набранных командами очков позволяет определить победителя, разница забитых и пропущенных мячей — разрешить неоднозначности при подведении итогов, количество и динамика количества побед и поражений той или иной команды — определить качество подготовки игроков в разные периоды турнира. Вам предлагается написать программу, считающую эту статистику.

Вам необходимо по данному списку команд, участвовавших в турнире, и списку игр, которые они провели между собой, определить количество забитых и пропущенных каждой командой мячей.

Формат входного файла

В первой строке входного файла **input.txt** находится одно натуральное число n ($1 \leq n \leq 10$) — количество команд, участвовавших в чемпионате. В следующих n строках перечислены названия этих команд. Каждое название команды находится в отдельной строке и является словом, состоящим только из строчных и заглавных букв латинского алфавита. Длина названия команды не превосходит 10, все названия команд различны.

В следующей строке входного файла находится одно целое число m ($1 \leq m \leq 100$) — количество матчей, которые команды сыграли за время чемпионата. Следующие m строк содержат результаты матчей.

Каждое описание матча содержится в отдельной строке. Описание матча состоит из названия первой команды, после которого через пробел следует счет матча. Счет матча — два числа, не превышающих 10, разделенные двоеточием. После счета матча через пробел дано название второй команды.

Гарантируется, что в матчах участвовали только команды, перечисленные в списке участников, и то, что команды, участвовавшие в матче, всегда различны.

Формат выходного файла

Выведите в выходной файл **output.txt** m строк, каждая из которых содержит название команды и количество забитых и пропущенных этой командой мячей. Между названием команды и количеством мячей должен стоять ровно один пробел, а между количеством забитых мячей и количеством пропущенных — ровно одно двоеточие. Команды перечислите в том же порядке, в котором они перечислены во входном файле.

Пример входных и выходных данных

input.txt	output.txt
3 CSKA Zenit Spartak 4 Zenit 3:2 CSKA Spartak 0:1 Zenit Spartak 1:1 CSKA CSKA 0:0 Zenit	CSKA 3:4 Zenit 4:2 Spartak 1:2

11. Технологии программирования (4 балла).

Сортировка чисел в массиве — одна из самых базовых задач в программировании. Разные сортировки отличаются друг от друга требуемым количеством памяти, временем работы, сложностью реализации и некоторыми менее важными параметрами.

Периодически возникают задачи, в которых массив необходимо отсортировать, используя некоторую нетривиальную операцию. Одну из таких задач вам и предлагается решить. В качестве сортирующей операции вы можете несколько раз использовать разворот последовательности из нескольких подряд идущих первых чисел массива. Например, вы можете превратить массив `[2, 3, 1, 4, 5]` в массив `[3, 2, 1, 4, 5]`, развернув первые два элемента, а затем превратить полученный массив в массив `[1, 2, 3, 4, 5]`, развернув первые три элемента.

Формат входного файла

В первой строке входного файла **input.txt** находится одно натуральное число n ($1 \leq n \leq 20$) — количество элементов в массиве. В следующей строке перечислены сами элементы массива — натуральные числа, не превышающие *1000*. Все элементы массива различны.

Формат выходного файла

В первой строке выходного файла **output.txt** требуется вывести одно целое число t , не превышающее *10000* — количество разворотов, которые вы собираетесь сделать. В следующей строке необходимо через пробел вывести t натуральных чисел, не превышающих n , где очередное число является количеством первых элементов массива, которые вы собираетесь развернуть соответствующей операцией. После выполнения выведенных вами операций массив должен быть отсортирован по возрастанию.

Пример входных и выходных данных

input.txt	output.txt
5 2 3 1 4 5	2 2 3