

**Отборочный этап 11 класса. 2 тур (приведен один из вариантов заданий)**

**1. Электронные таблицы. Адресация ячеек и вычисления (1 балл)**

**[100x100]**

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D
1	=СУММ(C3:CX102)			
2		0	=B2+\$B\$1	
3		=B2+\$A\$2	=ЕСЛИ(C\$2>\$B3;1;ЕСЛИ(C\$2<\$B3;-1;0))	
4				

Ячейку C2 скопировали во все ячейки диапазона D2:CX2. Ячейку B3 скопировали во все ячейки диапазона B4:B102. Ячейку C3 скопировали во все ячейки диапазона C3:CX102. В ячейку B1 поместили число 73. Какое целое положительное число необходимо поместить в ячейку A2, чтобы в ячейке A1 получилось значение 5127. В ответе укажите целое число.

Ответ: 36

## 2. Сортировка и фильтрация данных (3 балла)

[Купить или произвести]

База данных системы управления производством состоит из трех таблиц:

Стоимость приобретения деталей	
Название детали	Стоимость приобретения
Деталь 1	137
Деталь 2	214
Деталь 3	4043
Деталь 4	1343
Деталь 5	
Деталь 6	132
Деталь 7	4124
Деталь 8	111
Деталь 9	74

Стоимость сборки деталей	
Название детали	Стоимость сборки
Изделие	142
Деталь 3	63
Деталь 4	29
Деталь 7	38

Состав		
Название собираемой детали	Название требуемой детали	Количество требуемых деталей
Изделие	Деталь 3	2
Изделие	Деталь 7	3
Деталь 3	Деталь 1	1
Деталь 3	Деталь 4	3
Деталь 3	Деталь 6	4
Деталь 4	Деталь 5	4
Деталь 4	Деталь 9	6
Деталь 7	Деталь 2	5
Деталь 7	Деталь 4	2
Деталь 7	Деталь 8	3

Задачей производства является сборка Изделия. Изделие собирается из деталей. Некоторые детали могут быть только приобретены, а некоторые детали могут быть или приобретены или собраны из других деталей. В таблице «Стоимость приобретения деталей» указана стоимость каждой детали в случае её приобретения. В таблице «Состав» указано, сколько каких деталей требуется для сборки одной детали из тех, которые можно собрать, включая само Изделие. В таблице «Стоимость сборки» указана стоимость сборки одной детали, которую можно собрать. При составлении плана производства система для каждой детали, которую можно собрать или приобрести выбирает тот способ получения

детали, который будет дешевле. Стоимость детали в случае её сборки складывается из стоимости необходимого количества других деталей и стоимости сборки.

При какой максимальной стоимости «Деталь 5» можно получить Изделие, стоимость которого не превышает 17647?

В ответе укажите целое число.

**Ответ: 126**

### 3. Мультимедиа технологии (2 балла)

#### [Цветовой тон]

Одним из инструментов, позволяющих дать обобщенную характеристику цветов растрового изображения, является гистограмма. Она представляет собой график, по оси абсцисс которого отложены яркости каналов изображения (в нашем случае красного, зеленого и синего соответственно), а по оси ординат количество точек, имеющих такую яркость в выбранном канале.

Одним из способов цветокоррекции изображений является преобразование цветов точек в цветовом пространстве HSB (Hue, Saturation, Brightness).

Пусть дано растровое изображение, содержащее пиксели четырех цветов:  $(R=250, G=128, B=5)$ ,  $(R=250, G=5, B=5)$ ,  $(R=128, G=250, B=5)$  и  $(R=250, G=250, B=5)$ . Изображение содержит равное количество пикселей каждого из этих четырех цветов.

Ниже приведены гистограммы каналов RGB изображения до и после цветокоррекции. Известно, что в процессе цветокоррекции изображение перевели в цветовую модель HSB, изменили значение Hue на  $+X$  градусов и перевели обратно в цветовую модель RGB. Выберите из предложенных вариантов значение  $X$ , которое было использовано при цветокоррекции.



Варианты ответа:

1. 15
2. 30
3. 45
4. 60
5. 75
6. 90
7. 105
8. 120
9. 135
10. 150
11. 165
12. 180

**Ответ: 5**

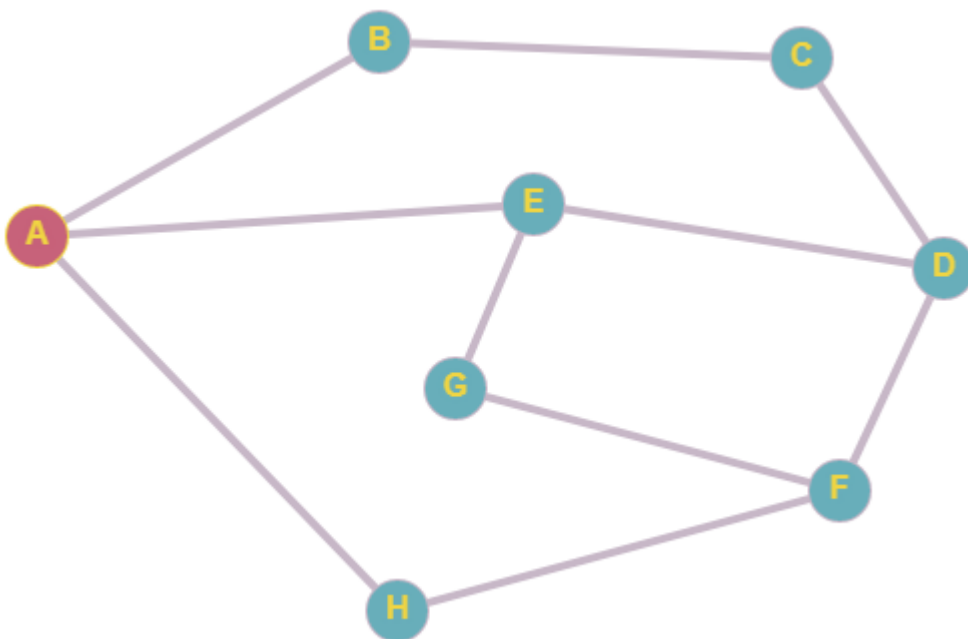
#### 4. Телекоммуникационные технологии (4 балла).

##### [Управление распределенной памятью]

Распределенные вычислительные системы часто используют для управления распределенной памятью метод «размножения страниц для чтения». В этом случае, если узлу распределенной вычислительной системы требуется страница памяти, которая есть на другом узле, и он планирует использовать её только для чтения данных, ему делается копия этой страницы. Соответственно, со временем, копии одной и той же страницы могут оказаться на значительном количестве узлов. Такой метод позволяет повысить производительность, но несет в себе риск нарушения целостности данных, если один из узлов решит поменять данные на странице. В этом случае для предотвращения появления различающихся по данным копий одной и той же страницы, требуется перед осуществлением модификации данных удалить все её копии в распределенной вычислительной системе. Для этого узел, который решит осуществить модификацию данных на странице, должен послать всем другим узлам уведомление об этом, дожидаясь подтверждения, что все копии удалены и только после этого начинать модификацию. Сложность в реализации такого решения заключается в том, что исходя из принципов построения распределенных вычислительных систем ни один узел не имеет прямого доступа ко всем остальным узлам и, тем более, не знает о наличии или отсутствии у них копий страниц. Узел может отправлять запросы и получать ответы только от нескольких соседних узлов распределенной вычислительной системы.

Рассмотрим модель.

Пусть есть распределенная вычислительная система из 8 узлов, связанных так, как показано на рисунке:



Известно время в секундах, которое требуется для передачи данных (запроса или отчета) от одного узла к другому:

	A	B	C	D	E	F	G	H
A	■	2			4			3
B	2	■	2					
C		2	■	4				
D			4	■	4	2		
E	4			4	■		2	
F				2		■	2	2
G					2	2	■	
H	3					2		■

Удаление копий реализуется в соответствии со следующими правилами:

1. Узел, который хочет модифицировать страницу, посылает запрос всем узлам, с которыми он связан, и ждет ответа. Запрос имеет поле "Путь". В это поле узел помещает список имен узлов, которым он направил запросы, и свое имя. Это делается для того, чтобы другие узлы не отправляли повторные запросы узлам, про которые уже известно, что им отправлены запросы.
2. Узел, который получил запрос или несколько запросов в определенный момент времени, выполняет следующие действия:
  - a. Получив все запросы, пришедшие в эту секунду, узел проверяет, есть ли связанные с ним узлы, которых нет в полях "Путь" полученных к этому моменту запросов. Если такие узлы есть, он посылает им запрос, в поле "Путь" которого указывает все узлы, которые содержатся в полях "Путь" полученных им запросов и дописывает своё имя и имена узлов, которым рассылает запросы. После этого он ждет от них отчетов.

- b. Если у узла, получившего запрос, есть страница, которую требуется удалить, узел начинает процедуру её удаления, которая занимает 4 секунды. Рассылка запросов и удаление страницы – независимые операции и начинаются одновременно.
3. Узел, получивший запрос, должен отправить отчет узлу, приславшему запрос. Он сможет это сделать, только когда он дождался отчетов от всех, кому сам послал запрос (если посылал) и когда удалил свою страницу (если она у него была). Если узлу не нужно было удалять страницу, и он не посылал запросы, он отправит ответ мгновенно. Если узел получил несколько запросов, как только у него будет готов отчет на первый из них, он вышлет его всем, от кого получил запросы (повторно запросы другим узлам он не посылает и удаление страницы не прерывает).
4. Если узел получает запрос от узла, которому ранее сам отправлял запрос, то, если у него нет страницы, которую нужно удалить, он мгновенно посылает отчет этому узлу. Если у него есть страница, дожидается её удаления и сразу посылает отчет этому узлу.

Через сколько секунд узел А, который решил модифицировать страницу, получит отчеты на все свои запросы, то есть сможет начать модификацию страницы, если известно, что копии страницы есть на узлах D, E и H? В ответе укажите целое число.

**Ответ: 23**

## 5. Операционные системы (2 балла)

### [Косвенная адресация]

Одним из распространенных способов организации файловой системы является использование индексных дескрипторов и многоуровневой адресации блоков данных.

Для каждого файла существует индексный дескриптор, который хранится **вне** области данных. Область данных разбита на блоки одинакового размера. Блоки пронумерованы последовательно натуральными числами. Индексный дескриптор хранит некоторые метаданные о файле и 15 номеров блоков из области данных.

Первые 12 номеров являются прямыми адресами – это номера первых 12 блоков с данными файла.

Если размер файла превышает 12 блоков, то 13-ый номер блока в индексном дескрипторе используется для косвенной адресации. Это значит, что он соответствует блоку **в области данных**, но содержимое этого блока будет интерпретировано не как продолжение данных файла, а как последовательность из 32-х битных чисел, являющихся номерами блоков данных, которые уже являются непосредственно данными файла.

Если размер файла настолько большой, что для его хранения не хватает блоков, которые адресуются прямыми адресами и косвенными адресами, используется 14-й номер блока в индексном дескрипторе, реализующий двойную косвенную адресацию. Это значит, что он соответствует блоку **в области данных**, содержимое которого интерпретируется как 32-х битные числа – номера блоков косвенной адресации, также расположенных **в области данных**, которые в свою очередь также состоят из 32-х битных чисел – уже номеров блоков данных файла.

Наконец, для самых больших файлов используется еще один уровень иерархии. 15-ый номер блока в индексном дескрипторе реализует тройную косвенную адресацию. Он соответствует блоку **в области данных**, содержимое которого интерпретируется как 32-х битные числа – номера блоков двойной косвенной адресации, расположенных **в области данных**, содержимое которых интерпретируется как 32-х битные числа – номера блоков косвенной адресации. Они также хранятся **в области данных** и уже их содержимое интерпретируется как 32-х битные числа – номера блоков данных файла.

Естественно, в зависимости от размера файла, количество блоков с адресами различных уровней, хранимых в области данных будет отличаться. При этом, единицей хранения всегда является блок. Поэтому даже, если необходимо хранить всего один адрес, который не поместился в предыдущий блок, будет использован целый блок, содержащий этот адрес.

Таким образом, при использовании многоуровневой адресации блоков данных все номера блоков, кроме 15, хранятся в блоках в области данных.

Пусть используются блоки размером 4 КБайта. Определите, сколько всего блоков будет занято в области данных для хранения файла размером в 17 ГБайт. В ответе укажите целое число.

Примечание:

1 ГБайт = 1024 МБайт; 1 Мбайт = 1024 КБайт; 1 КБайт = 1024 байт.

**Ответ: 4460806**

## 6. Технологии программирования (3 балла)

### [Подготовка к марафону]

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

Мальчик Саша готовится к марафону. Для того, чтобы понять какой результат он должен показать в этом году, Саша изучает результаты прошлого года.

Результаты представлены в виде таблицы, каждая строка таблицы содержит номер контрольной точки, имя участника и время прохождения точки. Первая контрольная точка является стартовой точкой, а последняя контрольная точка является точкой финиша. Все записи отсортированы в хронологическом порядке.

Участники могут стартовать в разное время, но весь марафон проходит в течение одних суток с 00:00:00 по 23:59:59. Не все участники приходят к финишу, некоторые могут сойти в середине дистанции или не успеть на финиш к концу марафона.

Саша хочет выступить не хуже первого участника прошлогоднего марафона. Помогите Саше узнать, какой участник пробежал быстрее всех. Если таких участников несколько, Саша хочет знать всех.

#### Формат входных данных

В первой строке даны два числа  $n$  и  $m$  - число контрольных точек и число записей в таблице результатов ( $2 \leq n \leq 50; 1 \leq m \leq 1000$ ).

Далее даны  $m$  строк таблицы результатов. В каждой строке через пробел заданы:

Номер контрольной точки  $p_i$  ( $1 \leq p_i \leq n$ ). Гарантируется, что все участники проходят точки последовательно без пропусков.

Имя участника, которое задается строкой из латинских букв, первая из которых заглавная, остальные строчные. Длина каждого имени не превосходит 20.

Время прохождения контрольной точки участником в формате hh:mm:ss. Гарантируется, что все записи времени корректны.

Также гарантируется, что все записи отсортированы в хронологическом порядке и каждый участник проходит одну контрольную точку не больше одного раза.

#### Формат выходных данных

В первой строке выведите число интересующих Сашу участников. Далее выведите имена всех интересующих Сашу участников, по одному в строке в лексикографическом порядке.

#### Примеры

Стандартный ввод	Стандартный вывод
3 7 1 Vasya 01:11:10 2 Vasya 01:22:23 1 Petya 12:00:00 3 Vasya 12:00:00 2 Petya 13:00:00 3 Petya 14:00:00 1 Kolya 23:59:59	1 Petya
2 4 1 Vasya 00:00:00 2 Vasya 01:00:00 1 Petya 22:00:00 2 Petya 23:00:00	2 Petya Vasya
3 4 1 Vasya 00:00:00 2 Vasya 01:00:00 1 Petya 22:00:00 2 Petya 23:00:00	0
2 6 1 Anya 00:00:00 2 Anya 01:00:00 1 Grisha 15:00:00 2 Grisha 16:00:00 1 Borya 22:00:00 2 Borya 23:00:00	3 Anya Borya Grisha

## 7. Технологии программирования (5 баллов)

### «Простая» задача

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

В данной задаче по данному массиву  $a$  нужно найти число способов выбрать четыре индекса  $i_1, j_1, i_2, j_2$  таких, что  $a_{i_1} \cdot a_{j_1} \leq a_{i_2} \cdot a_{j_2}$ .

Обратите внимание на ограничения: если просто перебрать числа циклами, вы получите превышение ограничения по времени!

**Формат входных данных**

В первой строке дано одно натуральное число  $n$  - число чисел в массиве ( $1 \leq n \leq 10^4$ ).

Во второй строке заданы  $n$  чисел  $a_0, a_1, \dots, a_{n-1}$  - элементы массива ( $1 \leq a_i \leq 500$ ).

**Формат выходных данных**

Выведите одно число - число способов выбрать четыре индекса  $i_1, j_1, i_2, j_2$  таких, что  $a_{i_1} \cdot a_{j_1} \leq a_{i_2} \cdot a_{j_2}$ .

**Примеры**

Стандартный ввод	Стандартный вывод
2 2 1	11
1 1	1
3 1 2 1	57
3 2 3 6	48

**Замечание**

В первом тесте под условие подходит 11 четверок:

№	$i_1$	$j_1$	$i_2$	$j_2$	$a_{i_1}$	$a_{j_1}$	$a_{i_2}$	$a_{j_2}$	$a_{i_1} \cdot a_{j_1}$	$a_{i_2} \cdot a_{j_2}$
1	0	0	0	0	2	2	2	2	4	4
2	0	1	0	0	2	1	2	2	2	4
3	0	1	0	1	2	1	2	1	2	2
4	0	1	1	0	2	1	1	2	2	2
5	1	0	0	0	1	2	2	2	2	4
6	1	0	0	1	1	2	2	1	2	2
7	1	0	1	0	1	2	1	2	2	2
8	1	1	0	0	1	1	2	2	1	4
9	1	1	0	1	1	1	2	1	1	2
10	1	1	1	0	1	1	1	2	1	2
11	1	1	1	1	1	1	1	1	1	1