

Отборочный этап 9 и 10 класса. 2 тур (приведен один из вариантов заданий)

1. Электронные таблицы. Адресация ячеек и вычисления (2 балла)

[25x25]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E
1	=СУММ(C3:AA27)				
2		0	1	=C2+2	
3		1	=ОСТАТ(C\$2;\$B3)+ОСТАТ(\$B3;C\$2)+B2		
4		=B3+2			
5					
6					

Ячейку D2 скопировали во все ячейки диапазона E2:AA2. Ячейку B4 скопировали во все ячейки диапазона B5:B27. Ячейку C3 скопировали во все ячейки диапазона C3:AA27. Затем удалили одну из ячеек диапазона C3:AA27 и получили в ячейке A1 значение 97088. Определите, какую ячейку удалили и запишите в ответ её адрес, например, C4. Если таких ячеек несколько, запишите в ответ адрес любой из них.

Ответ: P10 | J16

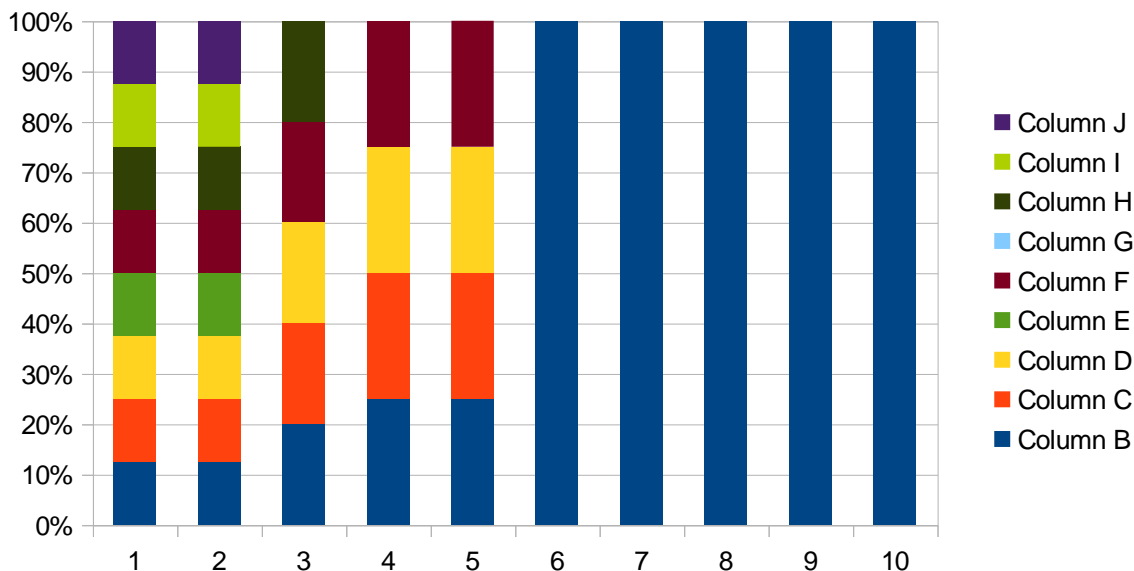
2. Электронные таблицы. Графики и диаграммы (1 балл)

[Остатки]

Дан фрагмент электронной таблицы в режиме отображения формул:

	A	B	C	D	E	F	G	H	I	J	K
1		2	3	4	5	6	7	8	9	10	
2	1	=ЕСЛИ(ОСТАТ(\$A\$1;СТЕПЕНЬ(B\$1;\$A2))=0;1;0)									
3	2										
4	3										
5	4										
6	5										
7	6										
8	7										
9	8										
10	9										
11	10										
12											

Ячейку B2 скопировали во все ячейки диапазона B2:J11. В ячейку A1 поместили целое положительное число N, выделили диапазон B2:J11 и построили нормированную гистограмму с накоплением. При каком минимальном значении N могла получиться такая диаграмма:



В ответе укажите целое число.

Ответ: 6220800

3. Сортировка и фильтрация данных (3 балла)

[Большой-меньший]

Петя придумал свой алгоритм сортировки одномерного массива. Он не очень эффективный, но Пете нравится.

Алгоритм состоит из следующих шагов:

1. Найти в массиве все пары элементов такие, что в них меньший по значению элемент имеет больший индекс, чем больший по значению элемент. Если таких пар нет, завершить исполнение алгоритма, иначе перейти к следующему шагу.
2. Из найденных пар элементов выбрать такую пару, разность между значениями которых будет максимальна. Если таких пар несколько, выбрать ту, у которой первый элемент имеет меньший индекс.
3. Поменять местами элементы в выбранной паре.
4. Перейти к шагу 1.

Петя взял массив [4,3,1,2] и использовал свой алгоритм. В результате он получил массив, отсортированный по возрастанию элементов. Петя обратил внимание, что шаг 3 его алгоритма в результате был исполнен 3 раза.

Тогда Петя решил взять массив с большим количеством элементов:

[10,16,6,13,11,18,8,5,9,0,2,15,4,3,7,14,12,19,1,17]

Сколько раз будет исполнен шаг 3 алгоритма Пети в этом случае?

В ответе укажите целое число.

Ответ: 38

4. Поиск и фильтрация данных (1 балл)

[Олимпиада по программированию]

В базе данных хранятся результаты олимпиады по программированию. Во время прохождения олимпиады каждый участник выполнял все задания на одном и том же языке программирования. Можно было использовать только языки программирования Python, C++, Java, Pascal. Петя сделал запросы к базе данных и получил следующие соотношения в количестве выдаваемых записей:

1. Количество записей, полученных по запросу «Участник выполнял свои задания на языке Python», ровно в 2 раза больше, чем количество записей, полученных по запросу «Участник выполнял задания на языке C++».
2. Количество записей, полученных по запросу «Участник выполнял свои задания на языке Java», ровно в 5 раз меньше, чем количество записей, полученных по запросу «Участник выполнял свои задания на языках Python или C++».
3. Количество записей, полученных по запросу «Участник выполнял свои задания на языке Pascal», составляет ровно X процентов от количества записей, полученных по запросу «Участник выполнял задания на языках C++ или Java», где X – целое число.

Как опытный участник олимпиад, Петя знает, что количество победителей не может превышать 8 процентов от количества участников, а суммарное количество победителей и призеров не может превышать 25 процентов от количества участников.

При каком минимальном X может быть истинным утверждение «Все победители выполняли свои задания на языке Pascal»?

В ответе укажите целое число.

Ответ: 20

5. Телекоммуникационные технологии (3 балла).

[Конкуренция за канал]

Два приложения (A и B) используют один канал связи для передачи данных. Балансировщик нагрузки на канал работает таким образом, что, если в данный момент времени передает данные только одно приложение, скорость передачи данных этого приложения составляет 96 МБит в секунду. Если одновременно передают данные два приложения, то канал делится пополам, и скорость передачи данных каждого приложения составляет 48 МБит в секунду. Изменения скорости могут происходить не чаще, чем раз в секунду, то есть если два приложения передавали данные одновременно, но одно из них закончило передачу до истечения очередной секунды, скорость передачи данных второго приложения не изменится до наступления новой секунды.

Приложения передают данные блоками через равные промежутки времени. Приложение A начинает передавать первый блок данных, размером в 96 МБайт в начальный момент времени и далее передает очередной блок данных такого же размера по истечении каждой 16-ой секунды. Приложение B начинает передавать первый блок данных, размером в 60 МБайт по истечении 4-ой секунды от начального момента времени и далее передает очередной блок такого же размера по истечении каждой 9-ой секунды от момента начала передачи своего первого блока.

Приложения не используют буферизацию, поэтому если получается ситуация, при которой одному из приложений нужно начать передачу своего очередного блока данных, но еще не закончилась передача предыдущего блока данных **этого** приложения, то происходит аварийная ситуация, передача данных обоими приложениями мгновенно останавливается и более не возобновляется. Определите, сколько блоков с данными успеет передать приложение A и сколько приложение B до возникновения аварийной ситуации. В ответе укажите через пробел два целых числа: сначала количество успешно переданных блоков приложения A и затем количество успешно переданных блоков приложения B. Если аварийной ситуации при заданных параметрах возникнуть не может, в ответе напишите NULL.

Ответ: 3 4

6. Операционные системы (2 балла)

[Shortest Job First]

Одним из алгоритмов планирования использования времени процессора в операционных системах является алгоритм Shortest Job First. Его принцип работы заключается в том, чтобы давать приоритет исполнению тому процессу, у которого наименьшее оставшееся время выполнения. Тогда "легкие" процессы будут быстрее покидать очередь к процессору и, тем самым, освобождать ресурсы вычислительного узла.

Рассмотрим модель работы планировщика. Время модели измеряется в условных тактах. Нумерация тактов идет с 1. Пусть для 5 процессов известны следующие два параметра: t_start - номер такта, в который данный процесс становится готов к исполнению и sru_burst - количество тактов процессорного времени, которые нужно предоставить процессу, чтобы он успешно исполнился. Эти данные приведены в таблице.

Процесс	t_start	sru_burst
P1	1	13
P2	4	6
P3	7	3
P4	1	11
P5	16	4

Планировщик предоставляет возможность исполняться на процессоре очередному процессу в течение кванта непрерывного исполнения, равного 4 тактам. По истечении каждого кванта непрерывного исполнения или в случае, когда очередной процесс завершил исполнение до окончания выделенного ему кванта непрерывного исполнения, планировщик выбирает следующий процесс. Планировщик всегда выбирает тот процесс, у которого остался минимальный среди всех

готовых к исполнению в данный такт процессов остаток от его сru-burst и предоставляет ему новый квант непрерывного исполнения. Если в момент принятия решения планировщиком существует несколько процессов с одинаковым минимальным остатком сru-burst, будет выбран процесс с меньшим порядковым номером. Если минимальный остаток сru-burst получается у процесса, который выполнялся в предыдущем кванте непрерывного выполнения, планировщик снова выберет его для исполнения. Считается, что на само принятие решений планировщиком не тратится время и оно принимается мгновенно.

Определите, когда закончит исполняться каждый из указанных в таблице процессов.

В ответе укажите через пробел 5 целых положительных чисел – номера тактов, в которые завершат работу процессы в порядке возрастания их порядковых номеров.

Ответ: 37 10 13 20 24

7. Технологии программирования (2 балла) [Техническое обслуживание]

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

В стране Точляндии все жители ездят на одинаковых автомобилях. Несмотря на то, что жители Точляндии выполняют техническое обслуживание точно в поставленный срок и не километром позже, многие из них постоянно забывают какие именно запчасти меняются в их автомобиле, но при этом очень любопытные и постоянно звонят на горячую линию, чтобы узнать эту информацию.

Техническое обслуживание в автомобилях Точляндии необходимо выполнять каждые 15000 километров. При этом для каждой запчасти известен срок ее службы. Так, например, масло требуется менять каждые 15000 километров, поэтому его меняют на каждом техническом обслуживании. А если срок службы запчасти 75000, ее будут менять только при пробегах 75000, 150000, 225000, 300000

Вас наняли в информационный отдел единой автомобильной компании Точляндии и просят реализовать сервис, который по пробегу автомобиля сможет определять список запчастей, которые менялись на предыдущем техническом обслуживании.

Формат входных данных

В первой строке даны два натуральных числа n и m - число сменных запчастей в автомобиле Точляндии и текущий пробег автомобиля ($1 \leq n \leq 100, 15000 < m < 1000000$). Гарантируется, что текущий пробег не является пробегом, на котором делают техническое обслуживание.

В следующих n строках даны запчасти и срок их службы в километрах. Название каждой запчасти состоит из не более чем двадцати строчных латинских букв. Гарантируется, что все названия различны. Срок службы - натуральное число кратное 15000, но не превосходящее 1000000.

Формат выходных данных

В первую строку выведите число k - число запчастей, которые менялись на предыдущем техническом обслуживании.

В следующих k строках выведите названия запчастей в лексикографическом порядке по одному названию на каждой строке.

Примеры

стандартный ввод	стандартный вывод
5 123567 interiorfilter 15000 sparkplugs 30000 oilfilter 15000 oil 15000 timingbelt 45000	4 interiorfilter oil oilfilter sparkplugs
1 100001 fullcar 990000	0

Замечание

При пробеге 123456 километров предыдущее техническое обслуживание было на пробеге 120000 километров, а следующее будет при пробеге 135000 километров.

8. Технологии программирования (4 балла) [Взлом терминала]

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

В этой задаче вам нужно взломать терминал. У терминала есть n числовых полей, чтобы его взломать необходимо, чтобы числа в полях являлись перестановкой чисел от 1 до n .

С терминалом можно делать следующие операции:

- Увеличить значение в любом числовом поле на единицу.
- Уменьшить значение в любом числовом поле на единицу.

Очевидно, что взломать терминал - это вопрос времени, но вам нужно сделать это как можно быстрее. По начальным значениям числовых полей определите минимальное число операций, необходимое, чтобы взломать терминал.

Формат входных данных

В первой строке дано одно натуральное число n - число числовых полей терминала ($1 \leq n \leq 1000$).

Во второй строке заданы n чисел a_1, a_2, \dots, a_n - начальные значения числовых полей терминала ($-1000 \leq a_i \leq 1000$).

Формат выходных данных

Выведите одно число - минимальное число операций, необходимое, чтобы взломать терминал.

Примеры

стандартный ввод	стандартный вывод
3 1 2 3	0
3 1 3 2	0
3 -1 -2 -3	12
3 -3 -2 -1	12
2 0 0	3