



Мы рады приветствовать участников
2 тура Олимпиады по информатике в ВГПУ
(направление «Программирование»)

Каждая задача имеет некоторую формальную постановку, в которой необходимо выделить алгоритмическую суть, т.е. пройти обязательный этап формализации в решении задачи. Для достижения успеха рекомендуем следовать совету Д. Пойа «Изучайте условие задачи пока вы окончательно не поймете, что именно вам нужно сделать». Когда дойдете до алгоритмической сути, то увидите, что задача очень проста.

Стоимость каждой задачи 2 тура Олимпиады указана в ее заголовке. Помните, что время Олимпиады ограничено и ваша задача заключается в том, чтобы набрать максимально возможное количество баллов, а не в том, чтобы полностью выполнить все задания. Иногда частичные решения нескольких заданий могут дать больше баллов, чем полное решение только одного. Разумно распределяйте свои силы и время.

Каждое задание должно быть сохранено в папку с номером задания. Если какое-то задание не выполнялось, то и папку с его номером создавать ненужно. В ряде заданий есть возможность представить несколько решений. В таком случае в папке должен быть файл с описанием того, какое решение к какой части задания относится. Файл должен называться `readme.txt`. После завершения Олимпиады сформируйте архив со всеми своими решениями и отправьте архив на почту nik.virt.pascal@gmail.com. Проверка работ будет осуществляться из числа поступивших на указанный почтовый адрес. Имя архива должно включать в себя номер компьютера, за которым выполнялась Олимпиада, и ФИО участника. Например, «422_05_Иванов_Константин.zip». В теме письма укажите Олимпиада по программированию в ВГПУ

Желаем успехов в решении задач!

Оргкомитет Олимпиады по информатике



01. Финансовый гений (100 + 70)



Всегда легко говорить о том, как нужно было поступать в прошлом, обладая всей полной информацией о произошедшем. Особенно часто такие «правильные советы» дают финансовые аналитики об использовании различных финансовых инструментов. Например, правильный совет заключается в том, чтобы покупать валюту тогда, когда она стоит мало, а продавать – когда она стоит

много. Проблема только в том, что в реальной жизни нельзя вернуться в прошлые дни и купить или продать валюту с учетом того, как она изменилась на сегодня, также нельзя заглянуть в курс валюты завтрашнего дня, чтобы понять свою стратегию сегодня. Но то, чего не знает человек, может знать программа, прочитав курсы валюты из файла.

Программа должна формировать стратегию по максимизации доходности валютных операций по известным значениям курса валюты при минимизации операций купли-продажи, т.е. зная курсы в определенные дни программа должна выдавать совет какую операцию в какой день следует выполнить, стараясь при этом не делать лишних операций.

Известно, что в начальный момент времени пользователь располагает суммой в A руб. Деньги в рублях ему потребуются снова через N дней. При этом известны значения курса валюты на каждый из этих дней. Будем считать, что в течение дня курс не меняется, а операции выполняются на всю имеющуюся в наличии сумму. Значение курса валюты, например, 12.30 означает, что нужно потратить 12.30 руб. для того, чтобы купить 1 условную единицу валюты или же при продаже 1 условной единицы валюты будет получено 12.30 руб.

Исходные данные (файл input.txt). В первой строке файла указаны 2 числа через пробел: начальная сумма в рублях A , которой располагает пользователь, а также количество дней N , в которые можно выполнять операции. В последующих N строках находится по одному вещественному числу – курс валюты на соответствующий день.

Результирующие данные (файл output.txt). В первой строке файла число – максимально возможный доход, который можно получить в результате валютных операций. В остальных строках должны быть указаны номер дня i , через пробел, тип операции (покупка валюты – buy или продажа валюты – sell). Если одинаковый максимальный результат можно получить разными способами, то следует выбрать тот, при котором будет наименьшее количество операций купли-продажи.



Пример

<i>input.txt</i>	<i>output.txt</i>
147 4 12.5 10.5 10.8 11.2	156.8 2 buy 4 sell
147 3 12.5 11.2 10.5	147

Бонусы: еще 70 баллов дополнительно к основному решению можно получить если в решении учесть дополнительное ограничение, что на k день пользователь должен иметь в рублях сумму не менее B руб.

В таком случае формат исходного файла содержит сначала строку с числами k и B , а затем полностью соответствует формату исходного файла основной задачи. Результирующий файл имеет такой же формат, что и в основной задаче.



02. Двоичная магия (40 | 100)



Петя заметил, что если перевести число в двоичную систему счисления, а затем переставить в нем разряды, то после обратного перевода в десятичную систему можно получить число как больше исходного, так и меньше исходного. Петя задался вопросом – какое максимальное и какое минимальное числа можно получить в результате таких перестановок для исходного числа N .

Этим открытием он поделился с Ваней. Ваня предложил еще более интересный пример – рассматривать не одно число, а несколько. При этом в двоичном представлении числа должны быть записаны в одинаковое количество разрядов. При необходимости добавляется нужное количество 0 в начало «коротких» чисел. Перестановка разрядов в двоичном представлении выполняется для всех рассматриваемых чисел одновременно. Например, если переставляется 1 и 2 разряды, то они переставляются у всех чисел сразу. После выполнения перестановок Ваня решил оценивать не отдельные числа, а их сумму, т.е. переводить в десятичную систему счисления сумму новых двоичных представлений.

Напишите программу, которая по описанному Ваней алгоритму определяет какая минимальная и максимальная суммы чисел могут быть получены.

Исходные данные (файл input.txt). В первой строке файла указано число анализируемых чисел N . В следующих N строках указаны сами числа. Числа не превосходят 1000.

Результирующие данные (файл output.txt). В первой строке файла указывается минимально возможная сумма чисел после преобразований, а во второй строке – максимально возможная сумма.

Пример

<i>input.txt</i>	<i>output.txt</i>
3	17
5	31
9	
3	
3	9
6	30
2	
10	



Примечание: Если напишите программу только для примера, предложенного Петей, то сможете получить всего 40 баллов. Входной файл при этом содержит только одно анализируемое число, а результирующий файл соответствует формату основной задачи.



03. Атипичная задача (100 + 10)



Обычно задачи на олимпиадах по программированию проверяются путем тестирования на некоторых наборах данных. Однако эта задача будет проверяться путем просмотра листинга программы, т.к. задание заключается в том, чтобы переписать код программы с некоторого несуществующего языка

программирования на ваш любимый язык. Листинг программы:

```
ЭТО A :: массив(10)
  i, m, nm :: число
##
измени i от 1 до 10
  прочитай A(i)

пусть m = A(1), nm = A(1)
измени i от 1 до 10
  если A(i) < m тогда пусть nm = m, m = A(i)
  иначе
    если A(i) < nm тогда пусть nm = A(i)

напиши nm
##
```

В первой строке программы создайте комментарий по правилам вашего языка программирования, в котором укажите его название. Далее запишите листинг, соответствующий заданию. Даже если вы поняли, как можно оптимизировать решение данной задачи, делать это не нужно. Ваш код должен полностью с точностью до синтаксиса соответствовать коду в условии задания.

Примечание: Еще 10 баллов можно получить, если в начале программы в комментарии указать что делает этот программный код.



04. Грузи – не тормози! (100)



Каждый из нас сталкивался с проблемой загрузки файла (страницы в интернет / мобильном приложении и т.д.) при низкой скорости канала связи. В таком случае отображается специальный значок – лоадер. Внешний вид бывает разным. В качестве примера мы выбрали такой, в котором на большой окружности размещены маленькие круги одинакового размера. Для простоты будем считать, что они практически не имеют зазора между собой.

Вам необходимо написать программу на любом языке программирования, которая подготавливает рисование такого лоадера. Параметрами программы являются радиус большой окружности R_0 и количество маленьких кругов N , которые должны быть изображены вдоль этой окружности. Так как рисование осуществляется пиксельное, то параметры малых кругов – координаты центров и радиусы, должны быть целыми числами. Если получается вещественный результат, то округлите его по математическим правилам. Будем считать, что размер экрана составляет 1200×800 точек, а начало координат находится в левом верхнем углу. Центр первого круга должен размещаться в самой высокой точке большой окружности (как 12 часов на циферблате), а остальные – по ходу часовой стрелки относительно первого.

Исходные данные (файл input.txt). В единственной строке файла через пробел указаны 2 числа, сначала R_0 , затем N . R_0 – радиус большой окружности, N – количество малых кругов, которые нужно разместить вдоль большой окружности.

Результирующие данные (файл output.txt). Файл должен состоять из N строк. В каждой строке через пробел должны быть указаны 3 целых числа X , Y , R . Каждая тройка чисел задает параметры одного круга. Порядок кругов при выводе в файл должен соответствовать описанному в задаче.

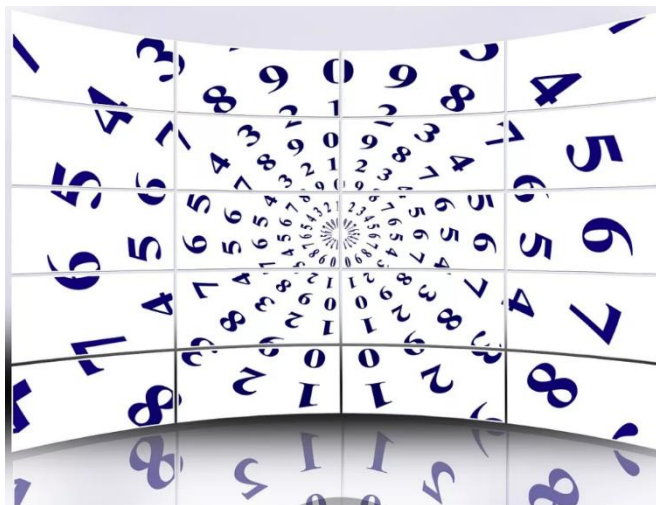


Пример

<i>input.txt</i>	<i>output.txt</i>
200 4	600 200 141 800 400 141 600 600 141 400 400 141
200 6	600 200 99 773 300 99 773 499 99 600 600 99 427 500 99 427 301 99



05. Нумерология (100)



Конечно же с научной точки зрения числа не имеют значения сами по себе и не влияют на жизнь человека. Следовательно, **нумерология это суеверие и лженаука**. Однако, Иван обратил внимание, что если он при выходе из дома встречает «счастливое» число, то день проходит успешно. Иван может увидеть это число как номер первого встреченного автомобиля или маршрутки, в которой едет в школу,

номер телефона на рекламном плакате и т.д. Т.к. Иван позитивный человек, то любое число пытается сделать «счастливым». Для этого он разбивает число на отдельные цифры, с которыми потом выполняет **операции сложения и вычитания**. В манипуляциях обязаны принимать участие все цифры исходного числа, но в произвольном порядке. Если после манипуляций получается число, совпадающее с возрастом Ивана, то такое число и считается «счастливым».

Напишите программу, которая по числу, которое встретил Иван, и его возрасту определяет будет ли день успешным. Если да, то запишите это выражение, т.е. какие операции и в каком порядке нужно выполнить над цифрами исходного числа чтобы получить возраст Ивана. Если это можно сделать разными способами, то запишите любой из них. Если же нельзя свести число к «счастливому», то в качестве результата выведите «Нумерология – это лженаука»

Исходные данные (файл input.txt). В единственной строке файла через пробел указаны 2 натуральных числа: сначала N, затем T. N – число, которое встретил Иван, T – возраст Ивана. Числа не превосходят 10^{30} . Числа могут быть действительно большими поскольку Иван может считать свой возраст не только в годах, но и месяцах, днях, часах и т.д.

Результирующие данные (файл output.txt). Файл должен состоять либо из строки «нумерология – это лженаука», либо из 2 строк: в первой строке – фраза «успешный день», а во второй – выражение из цифр встреченного числа (N), равное возрасту Ивана (T).



Пример:

<i>input.txt</i>	<i>output.txt</i>
152 16	нумерология – это лженаука
5271 13	успешный день 5+2+7-1
2837 16	успешный день 3+7-2+8