

## Лабораторная работа №1

### Вариант 1

Как уже несомненно стало известно всем, Запорожье добился права на проведение Летних Олимпийских игр 2032 года. Планируется, что Украине, как стране-организатору Олимпийских игр, будет разрешено внести минимальные изменения в программу Олимпиады. Так, с целью улучшения общего командного результата, было решено заменить соревнования по плаванию первенством в абсолютно новой игре "Пуговицы".

Правила игры очень просты. Перед двумя играющими находится кучка из  $K$  пуговиц. Играющие по очереди берут пуговицы из кучки, причем за один ход каждый из них может взять от 1 до  $L$  пуговиц. Выигрывает тот из спортсменов, которому удастся взять последнюю пуговицу.

Правила олимпийских соревнований будут лишь немного сложнее обычных. Тот из игроков, которому по жребию выпадает делать первый ход, получает возможность собственноручно назначить число  $K$ , руководствуясь в своем выборе только ограничениями  $3 \leq K \leq 100\,000\,000$  (именно столько пуговиц заготовлено для олимпийского турнира). Тот из игроков, который будет ходить вторым, выбирает, в свою очередь, число  $L$ , которое должно отвечать условию  $2 \leq L < K$ .

Задача

На вашу команду возлагается очень ответственная задача: необходимо написать программу, которая помогала бы второму игроку делать свой выбор. Другими словами, по заданному числу пуговиц в кучке  $K$ , необходимо определить такое число  $L$ , которое гарантирует победу второму игроку при наилучшей игре обеих сторон.

Так, например, если в кучке всего три пуговицы, то победу второму игроку обеспечивает выбор  $L=2$ . В самом деле, если первый игрок своим ходом заберет одну пуговицу, то второй сможет выиграть, взяв обе оставшихся пуговицы и, напротив, если первый возьмет две пуговицы, то второй победит, взяв последнюю. Исходные данные

Вход для этой задачи состоит из одной строки, в которой записано единственное число  $K$  - количество пуговиц в кучке, выбранное первым игроком.

Результат

На выход следует записать единственное целое число  $L$  - максимальное количество пуговиц, которое можно взять за один ход - обеспечивающее победу второму игроку. Если таких чисел несколько, то следует вывести наименьшее из них. Если таких чисел нет, то следует вывести число 0.

Пример исходных данных

3

Пример результата

2

### Вариант 2

В одном из островных государств Карибского бассейна все решения традиционно принимались простым большинством голосов на общем собрании граждан, которых, к счастью, было не очень много. Одна из местных партий, стремясь прийти к власти как можно более законным путем, смогла добиться некоторой реформы избирательной системы. Главным аргументом было то, что население острова в последнее время значительно возросло, и проведение общих собраний перестало быть легкой задачей.

Суть реформы состояла в следующем: с момента введения ее в действие все избиратели острова делились на  $K$  групп (необязательно равных по численности). Голосование по любому вопросу теперь следовало проводить отдельно в каждой группе, причем считалось, что группа высказывается "за", если "за" голосует более половины людей в этой группе, а в противном случае считалось, что группа высказывается "против". После

проведения голосования в группах подсчитывалось количество групп, высказавшихся "за" и "против", и решение вопрос решался положительно в том и только том случае, когда групп, высказавшихся "за", оказывалось более половины общего количества групп.

Эта система вначале была с радостью принята жителями острова. Когда первые восторги рассеялись, очевидны стали, однако, некоторые недостатки новой системы. Оказалось, что сторонники партии, предложившей систему, смогли оказать некоторое влияние на формирование групп избирателей. Благодаря этому, они получили возможность проводить некоторые решения, не обладая при этом реальным большинством голосов!

Пусть, например, на острове были сформированы три группы избирателей, численностью в 5, 5 и 7 человек соответственно. Тогда партии достаточно иметь трех сторонников в каждой из первых двух групп, и она сможет провести решение всего 6-ю голосами "за", вместо 9-и, необходимых при общем голосовании.

Вам надо написать программу, которая определяет по заданному разбиению избирателей на группы минимальное количество сторонников партии, достаточное, при некотором распределении их по группам, для принятия любого решения.

Ввод.

Входной файл для этой задачи состоит из двух строк. В первой строке файла записано натуральное число  $K < 101$  - количество групп избирателей. Во второй строке через пробел записаны  $K$  натуральных чисел, которые задают количество избирателей в группах. Для упрощения определения понятия "большинство голосов" будем полагать, что и число групп, и количество избирателей в каждой группе суть нечетные числа. Вы можете также считать, что население острова не превосходит 10001 человек.

Вывод.

В выходной файл следует записать единственное натуральное число - минимальное количество сторонников партии, способное решить исход голосования.

Пример файла input.txt.

3

5 7 5

Пример файла output.txt.

6

### Вариант 3.

Прямоугольники, площади которых равны, называются равновеликими. Написать программу, находящую все возможные целочисленные стороны равновеликих прямоугольников заданной площади. Одинаковые прямоугольники, получающиеся заменой сторон, печатать не надо.

Входные данные:

Площадь четырехугольника ( $1 \leq n \leq 50000$ ).

Выходные данные:

Стороны четырехугольника ( $A*B$ )

Пример входных данных:

12

Пример выходных данных:

1\*12 2\*6 3\*4

### Вариант 4

Имеется  $N$  банок с целочисленными объемами  $V_1, \dots, V_n$  литров, пустой сосуд и кран с водой. Можно ли с помощью этих банок налить в сосуд ровно  $V$  литров воды?

Исходная информация содержится в файле input.txt в первой строчке - количество банок  $N < 100$ , затем  $N$  строчек с объемами банок и в последней строчке объем сосуда.

Программа должна вывести в файл output.txt одно из двух слов YES! или NO.

## Вариант 5

Имеются монеты с различными фиксированными номиналами, выраженными в копейках (например, 3 и 5 копеек) в достаточном количестве. Написать программу COINS.\*, которая:

- а) определяет, можно ли представить заданную сумму  $S$  (выраженную в копейках), пользуясь монетами заданных номиналов,
- б) если это возможно, то представляет эту сумму с помощью минимального количества монет.

Входные данные: Входной файл COINS.DAT содержит в первой строке сумму  $S$  ( $0 < S < 1000000000$ ), во второй строке -  $N$  - количество различных номиналов ( $0 < N < 20$ ), а в следующих  $N$  строках -  $A_1 \dots A_N$  - номиналы (в порядке возрастания), которые можно использовать ( $0 < A_1 < A_2 < \dots < A_N < 1000000000$ ).

Выходные данные: Выходной файл COINS.SOL должен содержать в первой строке знак "+", если заданную сумму  $S$  можно представить, и знак "-", если нельзя. Если представленная сумма существует, то следующие  $N$  строк должны содержать количества монет каждого номинала, которые необходимы для представления суммы  $S$  с помощью минимального количества монет.

Примеры ввода и вывода

Представить 514 копеек с помощью монет номиналом в 3 и 5 копеек

COINS.DAT:

514

2

3

5

COINS.SOL:

+

3

101

Представить 27 копеек с помощью монет номиналом в 5 и 13 копеек

COINS.DAT:

27

2

5

13

COINS.SOL:

-

## Вариант 6

В детстве у меня было развлечение - кидаться помидорами с балкона, так чтобы забрызгать прогнившими внутренностями помидор прохожих. Я тогда подметил, что на каждый кубический сантиметр помидоры (кстати, она имеет форму идеального шара) приходится квадратный метр поверхности (это объясняется тем, что помидора падает с 7-го этажа и размазывается по большой площади мелкими брызгами). Т.е. помидора объемом в 3 куб. см. забрызгает круг с центром в точке падения площадью 3 кв. м.

Я наметил несколько точек, в которые я точно попаду. В какую из этих точек надо кинуть помидору наименьшего размера, чтобы забрызгать всех прохожих?

Входные данные:

В первой строке входного файла input.txt содержится число  $n$  - количество намеченных точек, в следующих  $n$  строках - координаты этих точек с точностью до 2 знаков после запятой. В следующей строке содержится  $m$  - количество человек, в последующих  $m$

строках - координаты людей.  $1 \leq m, n \leq 100$ . Координаты находятся в промежутке от -100 до 100.

Выходные данные:

В первой строке содержатся координаты точки, в которую надо кидать помидору, а во второй - наименьший радиус  $r$  помидоры в сантиметрах.

Пример входных данных:

```
1
0 0
2
1 0
0.5 0.5
```

Пример выходных данных:

```
0.00 0.00
0.91
```

### Вариант 7

В моей группе учится один парень - Ариф, который часто занимается на уроках математики в общем то математикой, но не совсем той, которой надо: то он выписывает в столбик квадраты всех чисел, то все степени двойки, то таблицу умножения до  $100 \times 100$ , короче говоря, наш Ариф - <кадр> еще тот. Недавно получил 2 за контрольную, потому что на всех предыдущих уроках занимался далеко не тем, чем надо. Однажды он придумал себе новое развлечение - стал выписывать в строку (без пробелов) четвертые степени всех простых чисел подряд. У него получилась примерно такая строка:

16816252401...

Потом ему стало интересно можно ли определить, какая цифра стоит на  $n$ -ом месте в этой строке? Формулу подобрать он не смог, да наверное ее и нету. Задача состоит в том, что несмотря на отсутствие формулы, определить, какая цифра стоит на  $n$ -ом месте.

Входные данные:

Во входном файле input.txt содержится единственное число  $n$  ( $0 < n < 25000$ ).

Выходные данные:

В выходном файле output.txt содержится одна цифра, которая стоит в этой строке на  $n$ -ом месте.

Пример 1:

input.txt:

1

output.txt:

1

Пример 2:

input.txt

5

output.txt

6

### Вариант 8

Жители одного государства очень любят различные математические головоломки. Даже тот, кто желает получить въездную визу, должен решить задачу: отыскать ключевое слово. Условие задачи таково:

На листке написано несколько длинных чисел. Если сложить все цифры в каждом числе, получатся новые числа. Далее, следует сложить все цифры в каждом из вновь полученных чисел. Процесс следует продолжать до тех пор, пока в результате не останутся числа, меньшие 10. После этого все просто: числа от 0 до 9 - это номера букв в алфавите (в этом

государстве алфавит состоит всего из десяти букв). Замена чисел буквами и дает ключевое слово.

Напишите программу, которая будет отыскивать ключевое слово.

Формат входного файла:

Первая строка - алфавит государства: десять букв расположенных по возрастанию порядковых номеров без пробелов.

Вторая строка - количество чисел ( $N \leq 255$ )

Следующие  $N$  строк - собственно исходные числа (по одному на строке, в каждом не более 255 цифр).

Формат выходного файла:

Ключевое слово

Пример:

input.txt

АГЕИКЛМОРТ

4

8267

19929

54262

000000000000

output.txt

ЛИГА

### Вариант 9

Написать программу, определяющую минимально возможное количество игроков в команде КВН, если известно, что девушек в команде больше  $X\%$ , но меньше  $Y\%$ .

Формат входных данных

В первой строке входного файла записаны числа  $X$  и  $Y$ , разделенные пробелом.  $X$  и  $Y$  - целые,  $1 \leq X, Y \leq 100$ .

Формат выходных данных

Нужно вывести одно число - минимальное количество членов команды при ограничениях, заданных во входном файле.

Пример ввода

40 50

Правильный вывод для примера

7

### Вариант 10.

Маршрут движения автомобиля задан в виде координат вершин ломаной.

Необходимо определить количество левых поворотов (смежные участки ломаной не лежат на одной прямой). Автомобиль начинает движение с любой точки (!!!Здесь уточнение от меня - автомобиль начинает движение из первой вершины ломаной, а вот ее координаты любые!!!).

Формат входных данных:

Первая строка входного файла input.txt состоит из одного числа, количества звеньев ломаной; в последующих строках - пары натуральных чисел, координаты вершин ломаной.

Формат выходных данных:

Выходной файл output.txt содержит одно число - количество левых поворотов

input.txt:

4

1 1

2 2

3 2  
3 3  
2 3  
output.txt:  
2

### Вариант 11.

На кольцевом маршруте №54 протяженностью  $S$ , проходящем мимо пансионата "Энергетик", работает  $N$  автобусов. Автобусы пронумерованы числами от 1 до  $N$  в порядке их следования по маршруту. Автобус с номером 1 движется за автобусом с номером  $N$ . Расписание составлено таким образом, что автобусы движутся с одинаковой скоростью  $V_0$  и с равными интервалами между ними. Движение автобусов контролирует диспетчер.

В 12 часов дня некоторые  $K$  автобусов одновременно снимаются с маршрута и отправляются на обед. Для восстановления равенства интервалов между автобусами, продолжающими движение по маршруту, потребуется некоторое время  $T$  и, возможно, изменение скорости некоторых автобусов по команде диспетчера. В течение этого времени автобусы должны двигаться с постоянными скоростями из интервала  $[V_{\min}, V_{\max}]$ , назначенными диспетчером. Изменение скорости движения автобуса происходит мгновенно. По истечении времени  $T$  автобусы возобновляют движение по маршруту со скоростью  $V_0$ .

Требуется написать программу для автоматического диспетчера, которая вычисляет минимальное время  $T_{\min}$ , за которое интервалы движения между оставшимися автобусами станут равными, и скорости движения каждого из них в течение этого времени.

Входные данные

Входной файл bus.in содержит две строки.

В первой строке находятся натуральные числа  $N$ ,  $K$ ,  $S$ ,  $V_{\min}$ ,  $V_{\max}$  и  $V_0$ , где  $K < N \leq 10000$ ,  $S \leq 10000$ ,  $V_{\min} < V_{\max} \leq 10000$ ,  $V_{\min} \leq V_0 \leq V_{\max}$ .

Во второй строке расположены в порядке возрастания  $K$  чисел - номера автобусов, снятых с маршрута. Все данные в строках разделены пробелами.

Выходные данные

В первой строке выходного файла bus.out должно находиться значение  $T_{\min}$ .

В каждой из последующих  $N - K$  строк должны быть по два разделенных пробелом числа - номер автобуса на маршруте и скорость его движения в течение времени  $T_{\min}$ . Номера автобусов упорядочить по возрастанию. Значения  $T_{\min}$  и скоростей выводить с точностью до 4-х значащих цифр после десятичной точки.

Пример 1

Входной файл bus.in

4 1 60 21 70 60

3

Выходной файл bus.out

0.2041

1 45.5

2 21

4 70

Пример 2

Входной файл bus.in

4 2 40 30 80 50

2 4

Выходной файл bus.out

0

1 50  
3 50

### Вариант 12.

В некотором царстве, в некотором государстве было  $N$  городов, и все они, судя по главной карте императора, имели целые координаты. В те годы леса были дремучие, дороги же строить умели только параллельно осям координат, так что расстояние между двумя городами определялось как  $|x_1 - x_2| + |y_1 - y_2|$ .

Император решил построить  $n+1$ -ый город и сделать его столицей своего государства, при этом координаты столицы также должны быть целыми. Место для столицы следует выбрать так, чтобы среднее арифметическое расстояний между столицей и остальными городами было как можно меньше. Однако, разумеется, столицу нельзя строить на месте существующего города.

Нелегкая задача выбрать место для столицы поручена Вам.

Формат входных данных

Первая строка входного файла содержит число  $N$  - количество городов ( $1 \leq N \leq 100$ ). Следующие  $N$  строк содержат координаты городов - пары целых чисел, не превышающих 1000 по абсолютной величине.

Формат выходных данных

Выведите в выходной файл два целых числа - координаты точки, где следует построить столицу. Если решений несколько, выведите любое.

Примеры

capital.in

8

0 0

1 0

2 0

0 1

2 1

0 2

1 2

2 2

capital.out

1 1

capital.in

4

0 0

1 1

0 1

1 0

capital.out

0 2

### Вариант 13.

Не секрет, что многие жители Лощины Янтарной Росы часто обращаются к монахам Монастыря с различными вопросами. Каждую весну жители окрестных деревень спрашивают Настоятеля, сколько мешков зерна они соберут осенью с одного посаженного весной мешка. Настоятелю надоело каждый раз изобретать ответ на этот вопрос, и он решил придумать алгоритм, который бы генерировал ответ (не подумайте, что он был просто шарлатаном, но нельзя же тревожить Будду по таким пустякам). Он решил, что будет отвечать по следующему правилу (такое уж хитрое правило никто не раскусит):

Факториалом числа  $n$ ,  $n$  - натуральное, Настоятель называет число, определяемое по правилу:  $n! = 1$ , если  $n = 1$

$n! = n \cdot (n - 1)!$ , если  $n > 1$

Функция  $F(x)$ ,  $x \geq 0$  вычисляется по следующему правилу:

$F(x) = x$ , если  $x \leq 9$

$F(x) = F(S(x))$ , если  $x > 9$

где  $S(x)$  - сумма цифр в десятичной записи числа  $x$ .

Правило это Настоятель применяет следующим образом: в год с номером  $n$  от рождения Будды мудрый Настоятель сообщает, что они соберут  $F(n!)$  мешков зерна с одного посаженного мешка весной. С каждым годом все трудней и трудней Настоятелю вычислять  $F(n!)$ , ваша цель - помочь ему в этом.

Ввод. Единственное число  $n$  - номер года,  $1 \leq n < 200$

Вывод.  $F(n!)$  - ответ Наставника

Пример input.txt #1

3

Пример output.txt #1

6

Пример input.txt #2

4

Пример output.txt #2

6

#### Вариант 14.

Легендарный учитель математики Юрий Петрович придумал забавную игру с числами. А именно, взяв произвольное целое число, он переводит его в двоичную систему счисления, получая некоторую последовательность из нулей и единиц, начинающуюся с единицы. (Например, десятичное число  $1910 = 1 \cdot 2^4 + 0 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0$  в двоичной системе запишется как 100112.) Затем учитель начинает сдвигать цифры полученного двоичного числа по циклу (так, что последняя цифра становится первой, а все остальные сдвигаются на одну позицию вправо), выписывая образуемые при этом последовательности из нулей и единиц в столбик - он подметил, что независимо от выбора исходного числа получающиеся последовательности начинают с некоторого момента повторяться. И, наконец, Юрий Петрович отыскивает максимальное из выписанных чисел и переводит его обратно в десятичную систему счисления, считая это число результатом проделанных манипуляций. Так, для числа 19 список последовательностей будет таким:

10011

11001

11100

01110

00111

10011

и результатом игры, следовательно, окажется число  $1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 28$ .

Поскольку придуманная игра с числами все больше занимает воображение учителя, отвлекая тем самым его от работы с ну очень одаренными школьниками, Вас просят написать программу, которая бы помогла Юрию Петровичу получать результат игры без утомительных ручных вычислений.

Формат входных данных

Входной файл содержит одно целое число  $N$  ( $0 \leq N \leq 32767$ ).

Формат выходных данных

Ваша программа должна вывести в выходной файл одно целое число, равное результату игры.



Пример  
d.in  
19  
d.out  
28

### **Вариант 15.**

Входной файл содержит набор слов (словом считается последовательность символов, не содержащая символа перевода строки). 1-я буква может быть как заглавной, так и строчной, остальные - строчные. Пусть некоторая последовательность букв встречается в обоих словах (регистр букв учитывается).

Процент совпадения назовем максимальной длину такой последовательности, деленную на длину более короткого из этих слов, умноженную на 100% и округленное до ближайшего целого. Требуется распределить слова по парам, так чтобы суммарный процент совпадения для всех пар был наибольшим.

Формат входного файла:

N  
слово1  
...  
словоN

где N - число слов от 2 до 12 (всегда четное). Длина каждого слова не превышает 20 символов.

Формат выходного файла:

слово слово процент\_совпадения  
...  
слово слово процент\_совпадения

где:

в каждой паре слова стоят в алфавитном порядке.

пары отсортированы в алфавитном порядке по первым словам в них.

Пример входного файла:

8  
Оля  
Ира  
Катя  
Наташа  
Сережа  
Илья  
Коля  
Саша

Файл результата для данного примера:

Илья Оля 67  
Ира Сережа 67  
Катя Коля 50  
Наташа Саша 75