

## Задача 1 Древняя Курская земля

Имя входного файла: Kursk.in  
Имя выходного файла: Kursk.out  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 Кб

На Курской земле люди живут уже с древних времен. Самые первые следы поселений эпохи верхнего палеолита относятся к XV - XX тысячелетию до нашей эры. В VI - V вв. до н.э. возникает при устье рек Кур и Тускарь первое городище - первая Курская крепость. Однако крепость Первокурска просуществовала недолго – люди ушли в связи с сарматским нашествием ко II в до н.э.

Во времена Киевской Руси Великий князь киевский строит новые крепости и восстанавливает старые. Так возникают города-крепости в Курском крае, которые в X-XII вв. приобретают все более пограничный, оборонительный характер. Они окружены дубовыми стенами. В это время появляются избы, кое-где и каменные дома.

Современные археологи при планировании очередных раскопок составили карту Курской области, на которую нанесли крепость Первокурска и город-крепость X века. В базу данных археологи занесли координаты левого нижнего угла ( $X_1, Y_1$ ) и длины сторон ( $H_1, W_1$ ) прямоугольного участка, на котором полностью разместилась крепость Первокурска. Аналогичным образом в базе данных сохранили информацию о крепости времен Киевской Руси ( $X_2, Y_2, H_2, W_2$  соответственно).

Для подсчета расходов на предстоящие летние раскопки археологи решили определить общую площадь работы.

**Требуется** написать программу, определяющую общую площадь раскопок, учитывая, что стороны занесенных в базу участков выделенных под раскопку каждой крепости, параллельны осям координат.

### Формат входных данных

В первой строке входного файла содержится четыре разделенных одним пробелом целых числа  $X_1, Y_1, H_1, W_1$  ( $-1000 \leq X_1 \leq 1000, -1000 \leq Y_1 \leq 1000, 1 \leq H_1 \leq 1000, 1 \leq W_1 \leq 1000$ ), описывающих прямоугольного участка, на котором полностью разместилась крепость Первокурска.

Во второй строке входного файла содержится четыре разделенных одним пробелом целых числа  $X_2, Y_2, H_2, W_2$  ( $-1000 \leq X_2 \leq 1000, -1000 \leq Y_2 \leq 1000, 1 \leq H_2 \leq 1000, 1 \leq W_2 \leq 1000$ ), описывающих прямоугольного участка, на котором полностью разместилась крепость времен Киевской Руси.

### Формат выходных данных

В выходной файл выведите одно целое число — длину забора общего прямоугольного забора.

### Примеры

Kursk.in	Kursk.out
-2 0 4 5 1 0 1 3	20
0 0 10 5 8 2 3 8	68

## Задача 2 English ЕГЭ

Имя входного файла: stat.in  
Имя выходного файла: stat.out  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 мегабайта

На ЕГЭ по английскому языку решили использовать возможности компьютерных технологий. Учащийся вводит свое сочинение в специальную компьютерную программу, которая сохраняет его в виде одной строки. Эксперт начинает проверку работы со статистического анализа текста. Он должен узнать, сколько слов в данной текстовой строке.

**Требуется** написать программу, помогающую эксперту определить, сколько слов в проверяемой строке текста. При этом следует учесть, что во строке текста, в которой могут встречаться:

- прописные и строчные (т.е. большие и маленькие) латинские буквы,
- пробелы,
- знаки препинания: точка, запятая, восклицательный и вопросительный знак,
- символ `-`, обозначающий в некоторых случаях тире, а в некоторых — дефис.

Слово — это последовательность подряд идущих латинских букв и знаков дефис, ограниченная с обоих концов. В качестве ограничителей могут выступать начало строки, конец строки, пробел, знак препинания, тире. Тире отличается от дефиса тем, что слева и справа от знака дефис пишутся буквы, а хотя бы с одной стороны от тире идет либо начало строки, либо конец строки, либо пробел, либо какой-либо знак препинания, либо еще одно тире.

### Формат входных данных

Во входном файле записана строка длиной не более 255 символов.

### Формат выходных данных

В выходной файл выведите одно число — количество слов, которые содержатся в исходной строке.

### Примеры

stat.in	stat.out
Hello , world!	2
www.olympiads.ru	3
Gyro-compass - this is a ...	4

## Задача 3 Игровой код

Имя входного файла: `gamecod.in`  
Имя выходного файла: `gamecod.out`  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 мегабайта

Игровой мир настолько богат и разнообразен, что создать полную классификацию его жанров пока никому не удалось. Тем не менее, одна из разновидностей игр под названием «квест» пользуется большой популярностью и привлекает многих любителей приключений.

В мире компьютерных игр под квестом понимают интерактивную историю, в которой игрок управляет главным героем и последовательно проходит сюжетную линию, выполняя различные задания. Основными элементами такой игры выступают тщательное исследование игрового мира и разрешение всевозможных головоломок. Важнейшая особенность квеста – невозможность двигаться дальше, не выполнив текущую задачу. Всю игру необходимо искать подсказки, коды, вещи, а затем решать – как и где их использовать

Любитель компьютерных квестов школьник Михаил на одном из этапов исследования игрового мира новейшего квеста `Original&KristiGame` понял принцип построения игровых кодов для решения предлагаемых ему головоломок.

Во-первых, найденное ранее целое десятичное число он должен переводить в двоичную систему счисления. Например, десятичное число  $21_{10} = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 1 \cdot 2^0$  в двоичной системе запишется как `101012`.

Во-вторых, Михаил должен выполнять циклический сдвиг вправо разрядов полученного двоичного числа. При этом последняя цифра становится первой, а все остальные сдвигаются вправо. Образующиеся при этом последовательности из нулей и единиц нужно выписывать в столбик и находить максимальное из записанных чисел. (Не стоит говорить о том, что Михаил конечно же сразу понял, что, начиная с некоторого момента получающиеся последовательности нулей и единиц повторяются!)

В-третьих, полученное максимальное число Михаил переводит обратно в десятичную систему счисления. Это и есть нужный для игры код. Например, для числа 21 список последовательностей будет таким:

```
10101
11010
01101
10110
01011
10101
```

...

Соответственно, результатом окажется число  $1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 26$ .

Поскольку этот процесс занимает большое время, отвлекая тем самым Михаила от игрового мира, от Вас **требуется** написать программу, которая помогла бы ему получать игровые кодовые числа 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 игровому коду.

### Пример

<code>gamecod.in</code>	<code>gamecod.in</code>
19	28

## Задача 4 Приз

Имя входного файла: prize.in  
Имя выходного файла: prize.out  
Максимальное время работы на одном тесте: 1 секунда  
Максимальный объем используемой памяти: 64 мегабайта

Летом многие школьники отправляются в лагеря за городом или на море, чтобы хорошо отдохнуть и поправить здоровье. Для них организуется много самых разнообразных мероприятий. И, конечно, самые любимые – это активные игры, причем связанные с поисками чего-нибудь ценного или вкусного.

Правила таких игр Вы, конечно, знаете. Игрокам дается карта или алгоритм прохождения пути к призу, например, такой: «Встаньте около одинокого дерева. Пройдите тридцать шагов в сторону парка, потом семнадцать шагов в сторону столовой, ..., наконец десять шагов в сторону большого булыжника. Приз находится под ним». Большая часть таких указаний просто сводится к прохождению какого-то количества шагов в одном из восьми направлений (1 — север, 2 — северо-восток, 3 — восток, 4 — юго-восток, 5 — юг, 6 — юго-запад, 7 — запад, 8 — северо-запад) (см. рис). Длина шага в любом направлении равна 1 м.

Путешествие по такому пути обычно является прекрасным способом посмотреть окрестности лагеря, однако все играют на время и ходят найти приз как можно быстрее. Поэтому игроки хотят идти напрямую в точку, где спрятан приз. Например, вместо того, чтобы проходить три шага на север, один шаг на восток, один шаг на север, три шага на восток, два шага на юг и один шаг на запад, можно пройти напрямую, использовав около 3.6 шага (см. рис. 1).

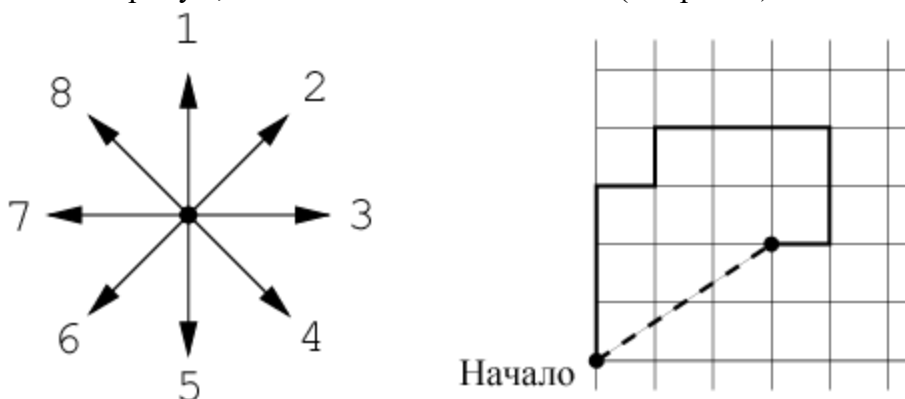


Рис. 1

От Вас требуется написать программу, которая по заданному алгоритму прохождения маршрута определяет точку, где спрятан приз.

### Формат входных данных

Первая строка входного файла содержит число  $N$  — число указаний ( $1 \leq N \leq 40$ ). Последующие  $N$  строк содержат сами указания — номер направления (целое число от 1 до 8) и количество шагов (целое число от 1 до 1000). Числа разделены пробелами.

### Формат выходных данных

В выходной файл выведите координаты  $X$  и  $Y$  точки (два вещественных числа, разделенные пробелом), где спрятан приз, считая, что ось  $Ox$  направлена на восток, а ось  $Oy$  — на север. В начале игрок должен стоять в начале координат. Координаты необходимо вывести с погрешностью не более  $10^{-3}$ .

### Примеры

prize.in	prize.out
6	3.000 2.000
1 3	
3 1	

Муниципальный этап Всероссийской олимпиады школьников по информатике и ИКТ 2017/18 уч.года  
Курская область, 10-11 классы

1 1 3 3 5 2 7 1	
1 8 10	-7.071 7.071